(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 840 512 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
06.05.1998 Bulletin 1998/19

(51) Int. Cl.[6]: **H04N 7/24**, H04N 5/00

(21) Application number: **97119123.4**

(22) Date of filing: **03.11.1997**

(84) Designated Contracting States:
AT BE CH DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE

(30) Priority: **01.11.1996 US 30106 P**

(71) Applicant:
TEXAS INSTRUMENTS INCORPORATED
Dallas, Texas 75243 (US)

(72) Inventors:
- **Chauval, Gerard**
  06600 Antibes (FR)
- **Benbassat, Gerard**
  06570 St Paul de Vence (FR)
- **Laczko, Frank L.**
  Allen, TX 75002 (US)
- **Chiang, Y. Paul**
  Richardson, TX 75082 (US)
- **Walker, Karen L.**
  Topeka, KA KS 66614 (US)

(74) Representative: **Holt, Michael**
Texas Instruments Limited,
Kempton Point,
68 Staines Road West
Sunbury-on-Thames, Middlesex TW16 7AX (GB)

(54) Integrated audio/video circuitry

(57) An improved audio-visual circuit is provided that includes a transport packet parsing circuit for receiving a transport data packet stream, a CPU circuit for initializing said integrated circuit and for processing portions of said data packet stream, a ROM circuit for storing data, a RAM circuit for storing data, an audio decoder circuit for decoding audio portions of said data packet stream, a video decoder circuit for decoding video portions of said data packet stream, an NTSC/PAL encoding circuit for encoding video portions of said data packet stream, an OSD coprocessor circuit for processing OSD portions of said data packets, a traffic controller circuit moving portions of said data packet stream between portions of said integrated circuit, an extension bus interface circuit, a P1394 interface circuit, a communication coprocessors circuit, an address bus connected to said circuits, and a data bus connected to said circuits.

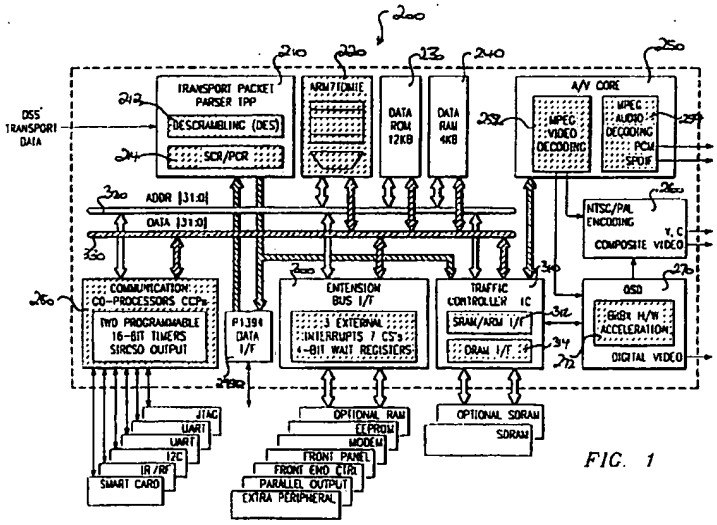


FIG. 1

EP 0 840 512 A2

Description

FIELD OF THE INVENTION

This invention generally relates to audio-visual systems and integrated circuits used therein, and more particularly, to improved audio-visual systems and improved integrated circuits used therein.

SUMMARY OF THE PRESENT INVENTION

Generally, and in one form of the present invention, an improved audio-visual circuit is provided that includes a transport packet parsing circuit for receiving a transport data packet stream, a CPU circuit for initializing said integrated circuit and for processing portions of said data packet stream, a ROM circuit for storing data, a RAM circuit for storing data, an audio decoder circuit for decoding audio portions of said data packet stream, a video decoder circuit for decoding video portions of said data packet stream, an NTSC/PAL encoding circuit for encoding video portions of said data packet stream, an OSD coprocessor circuit for processing OSD portions of said data packets, a traffic controller circuit for moving portions of said data packet stream between portions of said integrated circuit, an extension bus interface circuit, a P1394 interface circuit, a communication coprocessors circuit, an address bus connected to said circuits, and a data bus connected to said circuits.

The present invention provides a fully functional decoder using a single 16 Mbit external SDRAM.

The present invention provides a decoder that accepts transport bitstreams up to 40 Mbits per second.

The present invention provides an on-chip DES module for descrambling.

The present invention provides a video decoder that decodes MPEG-1 and MPEG-2 Main Profile and Main Level bitstreams.

The present invention provides an audio decoder that decodes MPEG-1 Layer I and II and MPEG-2 Multichannel bitstreams.

The present invention provides an audio output in both PCM and SPDIF formats.

The present invention provides an OSD processor that enables mixture of OSD and video data with transparent BitBLT hardware that accelerates memory block move.

The present invention provides a 32/16 bit ARM/Thumb processor that removes the need of another CPU in the set-top box.

The present invention provides firmware that controls device operation and provides application access to hardware resources.

The present invention provides an on-chip NTSC/PAL encoder that incorporates Closed Caption and Video Aspect Ratio Identification Signal encoding and the MacroVision logic for anti-taping protection.

The present invention provides an analog Y, C, and Composite video outputs with 9-bit precision.

The present invention provides an internally or externally generated sync signals.

The present invention provides a digital video component output that also contains Aspect Ratio ID.

The present invention provides an on-chip SDRAM controller for 16, 20, 24, or 32 Mbit SDRAM.

The present invention provides a general purpose 16-bit Extension Bus.

The present invention provides a 1394 interface that allows connection to external 1394 devices.

The present invention provides two 2-wire UART data ports.

The present invention provides a Smart Card interface.

The present invention provides an I^2C master/slave interface.

The present invention provides one IR, one SIRCSI, and one RF input data port.

The present invention provides two general purpose I/O pins.

The present invention provides a JTAG interface.

The present invention provides a 3.3 volt device with some 5 volt tolerant pins for interfacing to the 5 volt devices.

It is an object of the present invention to provide a fully functional decoder using a single 16 Mbit external SDRAM.

It is an object of the present invention to provide a decoder that accepts transport bitstreams up to 40 Mbits per second.

It is an object of the present invention to provide an on-chip DES module for descrambling.

It is an object of the present invention to provide a video decoder that decodes MPEG-1 and MPEG-2 Main Profile and Main Level bitstreams.

It is an object of the present invention to provide an audio decoder that decodes MPEG-1 Layer I and II and MPEG-2 Multichannel bitstreams.

It is an object of the present invention to provide an audio output in both PCM and SPDIF formats.

It is an object of the present invention to provide an OSD processor that enables mixture of OSD and video data with transparenct BitBLT hardware that accelerates memory block move.

It is an object of the present invention to provide a 32/16 bit ARM/Thumb processor that removes the need of another CPU in the set-top box.

It is an object of the present invention to provide a firmware that controls device operation and provides application access to hardware resources.

It is an object of the present invention to provide an on-chip NTSC/PAL encoder that incorporates Closed Caption and Video Aspect Ratio Identification Signal encoding and the MacroVision logic for anti-taping protection.

It is an object of the present invention to provide an analog Y, C, and Composite video outputs with 9-bit precision.

It is an object of the present invention to provide an internally or externally generated sync signals.

It is an object of the present invention to provide a digital video component output that also contains Aspect Ratio ID.

It is an object of the present invention to provide an on-chip SDRAM controller for 16, 20, 24, or 32 Mbit SDRAM.

It is an object of the present invention to provide a general purpose 16-bit Extension Bus.

It is an object of the present invention to provide a 1394 interface that allows connection to external 1394 devices.

It is an object of the present invention to provide two 2-wire UART data ports.

It is an object of the present invention to provide a Smart Card interface.

It is an object of the present invention to provide an I^2C master/slave interface.

It is an object of the present invention to provide one IR, one SIRCSI, and one RF input data port.

It is an object of the present invention to provide two general purpose I/O pins.

It is an object of the present invention to provide a JTAG interface.

It is an object of the present invention to provide a 3.3 volt device with some 5 volt tolerant pins for interfacing to the 5 volt devices.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention may be understood by referring to the detailed description which follows, read in conjunction with the accompanying drawings in which:

Figure 1 depicts a high level functional block diagram of a circuit that forms a portion of the audio-visual system of the present invention;

Figure 2 depicts a portion of and data flow between these portions of Figure 1.

Figure 3 shows an input timing for the circuit of Figure 1.

Figure 4 shows timing for the VARIS output.

Figure 5 shows timing for 4:2:2 and 4:4:4 digital video output;

Figure 6 depicts the data output of PCMOUT alternates between the two channels, as designated by LRCLK;

Figure 7 shows an example circuit where maximum clock jitter will not exceed 200 ps RMS;

Figure 8 (read) and Figure 9 (write) show Extension Bus read and write timing, both with two programmable wait states;

Figure 10 shows a timing diagram for a read operation with EXTWAIT signal on;

Figure 11 depicts a connection between the circuitry, an external packetizer, Link layer, and Physical layer devices;

Figure 12 shows a functional block diagram of data flow between the TPP, DES, and 1394 interfaces.

Figure 13 and Figure 14 depict read and write timing relationships on the 1394 interface;

Figure 15 shows a data path for an ARM processor core.

Figure 16 depicts the data flow managed by a Traffic Controller;

Figure 17 shows an example circuit for the external VCXO.

Figure 18 shows the block diagram of an OSD module;

Figure 19 shows example displays of two output channels;

Figure 20 show an example of the IR input bitstream; and

Figure 21 shows a model of the hardware interface.

Corresponding numerals and symbols in the different Figures refer to corresponding parts unless otherwise indicated.

DETAILED DESCRIPTION

Referring now to Figure 1 there may be seen a high level functional block diagram of a circuit 200 that forms a portion of an audio-visual system of the present invention and its interfaces with off-chip devices and/or circuitry. More particularly, there may be seen the overall functional architecture of a circuit including on-chip interconnections that is preferably implemented on a single chip as depicted by the dashed line portion of Figure 1.

As depicted inside the dashed line portion of Figure 1, this circuit consists of a transport packet parser (TPP) block 210 that includes a bitstream decoder or descrambler 212 and clock recovery circuitry 214, an ARM CPU block 220, a data ROM block 230, a data RAM block 240, an audio/video (A/V) core block 250 that includes an MPEG-2 audio decoder 254 and an MPEG-2 video decoder 252, an NTSC/ PAL video encoder block 260, an on screen display (OSD) controller block 270 to mix graphics and video that includes a bitblt hardware (H/W) accelerator 272, a communication co-processors (CCP) block 280 that includes connections for two UART serial data interfaces, infra red (IR) and radio frequency (RF) inputs, SIRCS input and output, an I2C port and a Smart Card interface, a P1394 interface (I/F) block 2990 for connection to an external 1394 device, an extension bus interface (I/F) block 300 to connect peripherals such as additional RS232 ports, display and control panels, external ROM, DRAM, or EEPROM memory, a modem and an extra peripheral, and a traffic controller (TC) block 310 that includes an SRAM/ARM interface (I/F) 312 and a DRAM I/F 314. There may also be seen an internal 32 bit address bus 320 that interconnects the blocks and an internal 32 bit data bus 330 that interconnects the blocks. External program and data memory expansion allows the circuit to support a wide range of audio/video systems, especially, for example, but not limited to, set-top boxes, from low end to high end.

The consolidation of all these functions onto a single chip with a large number of inputs and outputs allows for removal of excess circuitry and/or logic needed for control and/or communications when these functions are distributed among several chips and allows for simplification of the circuitry remaining after consolidation onto a single chip. More particularly, this consolidation results in the elimination of the need for an external CPU to control, or coordinate control, of all these functions. This results in a simpler and cost-reduced single chip implementation of the functionality currently available only by combining many different chips and/or by using special chipsets. However, this circuit, by its very function, requires a large number of inputs and outputs, entailing a high number of pins for the chip.

In addition, a JTAG block is depicted that allows for testing of this circuit using a standard JTAG interface that is interconnected with this JTAG block. As more fully described later herein, this circuit is fully JTAG compliant, with the exception of requiring external pull-up resistors on certain signal pins (not depicted) to permit 5v inputs for use in mixed voltage systems.

In addition, Figure 1 depicts that the circuit is interconnected to a plurality of other external blocks. More particularly, Figure 1 depicts a set of external memory blocks. Preferably, the external memory is SDRAM, although clearly, other types of RAM may be so employed. The external memory 300 is described more fully later herein. The incorporation of any or all of these external blocks and/or all or portions of the external memories onto the chip is contemplated by and within the scope of the present invention.

Referring now to Figure 2, it may be seen how the circuitry (`AV310) accepts a transport bitstream from the output of a Forward Error Correction (FEC) device with a maximum throughput of 40 Mbits/s or 7.5 Mbytes/s. The Transport Packet Parser (TPP) in the `AV310 processes the header of each packet and decides whether the packet should be discarded, further processed by ARM CPU, or if the packet only contains relevant data and needs to be stored without intervention from the ARM. The TPP sends all packets requiring further processing or containing relevant data to the internal RAM via the Traffic Controller (TC). The TPP also activates or deactivates the decryption engine (DES) based on the content of an individual packet. The conditional access keys are stored in RAM and managed by special firmware running on the ARM CPU. The data transfer from TPP to SRAM is done via DMA set up by the Traffic Controller (TC).

Further processing on the packet is done by the ARM firmware, which is activated by interrupt from the TPP after the completion of the packet data transfer. Two types of transport packets are stored in the RAM and managed as a first-in first-out (FIFO). One is for pure data which will be routed to SDRAM without intervention from the ARM, and the other is ford packets that need further processing. Within the interrupt service routine, the ARM checks the FIFO for packets that need further processing, performs necessary parsing, removes the header portion, and establishes DMA for transferring payload data from RAM to SDRAM. The Traffic Controller repacks the data and gets rid of the voids created by any header removal.

Together with the ARM, the TPP also handles System Clock Reference (SCR) recovery with an external VCXO. The TPP will latch and transfer to the ARM its internal system clock upon the arrival of any packet which may contain system clock information. After further processing on the packet and identifying the system clock, the ARM calculates the difference between the system clock from a bitstream and the actual system clock at the time the packet arrives. Then, the ARM filters the difference and sends it through a Sigma-Delta DAC in the TPP to control an external voltage controlled oscillator (VCXO). During start-up when there is no incoming SCR, the ARM will drive the VCXO to its center frequency.

The TPP will detect packets lost from the transport stream. With error concealment by the audio/video decoder and the redundant header from DSS bitstream, the `AV310 minimizes the effect of lost data.

After removing packet headers and other system related information, both audio and video data is stored in external SDRAM. The video and audio decoders then read the bitstream from SDRAM and process it according to the ISO standards. The chip decodes MPEG-1 and MPEG-2 main profile at main level for video and Layer I and II MPEG-1 and MPEG-2 for audio. Both Video and Audio decoders synchronize their presentation using the transmitted Presentation

Time Stamps (PTS). In a Digital Satellite System (DSS), the PTS is transmitted as picture user data in the video bitstream and an MPEG-1 system packet bitstream for audio. Dedicated hardware decodes the PTS if it is in the MPEG-1 system packet and forwards it to the audio decoder. The video decoder decodes PTS from picture user data. Both Video and Audio decoders compare PTS to the local system clock in order to synchronize presentation of reconstructed data. The local system clock is continuously updated by the ARM. That is, every time the System Clock Reference of a selected SCID is received and processed, the ARM will update the decoder system clock.

The Video decoder is capable of producing decimated pictures using 1/2 or 1/4 decimation per dimension, which results in reduced areas of 1/4 or 1/16. The decimated picture can be viewed in real time. Decimation is achieved by using field data out of a frame, skipping lines, and performing vertical filtering to smooth out the decimated image.

When decoding a picture from a digital recorder, the decoder can handle trick-modes (decode and display I frame only), with the limitation that the data has to be a whole picture instead of several intra slices. Random bits are allowed in between trick mode pictures. However, if the random bits emulate any start code, it will cause unpredictable decoding and display errors.

Closed Caption (CC) and Extended Data Services (EDS) are transmitted as picture layer user data. The video decoder extracts the CC and EDS information from the video bitstream and sends it to the NTSC/PAL encoder module.

The video decoder also extracts the aspect ratio from the bitstream and sends it to the ARM which prepares data according to the Video Aspect Ratio Identification Signal (VARIS) standard, EIAJ CPX - 1204. The ARM then sends it to the NTSC/PAL encoder and OSD module.

The OSD data may come from the user data in the bitstream or may be generated by the application executed on the ARM. Regardless of the source, the OSD data will be stored in the SDRAM and managed by the ARM. However, there is only limited space in the SDRAM for OSD. Applications that require large quantities of OSD data have to store them in an external memory attached to the Extension Bus. Based on the request from the application, the ARM will turn the OSD function "ON" and specify how and where the OSD will be mixed and displayed along with the normal video sequence. The OSD data can be represented in one of the following forms: bitmap, graphics 4:4:4 component, CCIR 601 4:2:2 component, or just background color. A special, dedicated bitBLT hardware expedites memory block moves between different OSDs.

The conditional access is triggered by the arrival of a Control Word Packet (CWP). The ARM firmware recognizes a CWP has been received and hands it to the Verifier, which is NewsDataCom (NDC) application software running on the ARM. The Verifier reads the CWP and communicates with the external Smart Card through a UART I/O interface. After verification, it passes the pointer to an 8 byte key back to the firmware, which then loads the key for the DES to decrypt succeeding packets.

The 32-bit ARM processor running at 40.5 MHz and its associated firmware provide the following: initialization and management of all hardware modules; service for selected interrupts generated by hardware modules and I/O ports; and application program interface (API) for users to develop their own applications.

All the firmware will be stored in the on-chip 12K bytes ROM, except the OSD graphics and some generic run time support. The 4.5K bytes on-chip RAM provides the space necessary for the `AV310 to properly decode transport bitstreams without losing any packets. The run-time support library (RTSL) and all user application software are located outside the `AV310. Details of the firmware and RTSL are provided in the companion software specification document.

There are two physical DMA channels managed by the Traffic Controller to facilitate large block transfers between memories and buffers. That is, as long as there is no collision in the source and destination, it is possible to have two concurrent DMA transfers. The detailed description of DMA is provided in the section on the Traffic Controller.

The `AV310 accepts DSS transport packet data from a front end such as a forward error correction (FEC) unit. The data is input 8 bits at a time, using a byte clock, DCLK. PACCLK high signals valid packet data. DERROR is used to indicate a packet that has data errors. The timing diagram in Figure 3 shows the input timing.

The `AV310 includes an interface to the Smart Card access control system. The interface consists of a high speed UART, logic to comply with the News Datacom specification (Document # HU-T052, Release E dated November 1994, and Release F dated January 1996) " Directv Project: Decoder-Smart Card Interface Requirements." Applicable software drivers that control the interface are also included, and are shown in the companion software document.

It should be noted that the `AV310 is a 3.3 volt device, while the Smart Card requires a 5 volt interface. The `AV310 will output control signals to turn the card's VCC and VPP "ON" and "OFF" as required, but external switching will be required. It is also possible that external level shifters may be needed on some of the logic signals.

A NTSC/PAL pin selects between an NTSC or a PAL output. Changing between NTSC and PAL mode requires a hardware reset of the device.

The `AV310 produces an analog S-video signal on two separate channels, the luminance (Y) and the chrominance (C). It also outputs the analog composite (Comp) signal. All three outputs conform to the RS170A standard.

The `AV310 also supports Closed Caption and Extended Data Services. The analog output transmits CC data as ASCII code during the twenty-first video line. The NTSC/PAL encoder module inserts VARIS codes into the 20th video line for NTSC and 23rd line for PAL.

The digital output provides video in either 4:4:4 or 4:2:2 component format, plus the aspect ratio VARIS code at the beginning of each video frame. The video output format is programmable by the user but defaults to 4:2:2. The content of the video could be either pure video or the blended combination of video and OSD.

The pin assignments for the digital video output signals are:

YCOUT(8) 8-bit Cb/Y/Cr/Y and VARIS multiplexed data output
YCCLK(1) 27 MHz or 40.5 MHz clock output
YCCTRL(2) 2-bit control signals to distinguish between Y/Cb/Cr components and VARIS code

The interpretation of YCCTRL is defined in the following table.

Table 1

Digital Output Control		
SIGNALS	YCCTRL[1]	YCCTRL[0]
Component Y	0	0
Component Cb	0	1
Component Cr	1	0
VARIS code	1	1

The aspect ratio VARIS code includes 14 bits of data plus a 6-bit CRC, to make a total of 20 bits. In NTSC the 14-bit data is specified as shown in Table 2.

Table 2

VARIS Code Specification		
	Bit Number	Contents
Word0 A	1	Communication aspect ratio: 1 = full mode (16:9), 0 = 4:3
	2	Picture display system: 1 = letter box, 0 = normal
	3	Not used
Word0 B	4 5 6	Identifying information for the picture and other signals (sound signals) that are related to the picture transmitted simultaneously
Word1	4-bit range	Identification code associated to Word0
Word2	4-bit range	Identification code associated to Word0 and other information

The 6-bit CRC is calculated, with the preset value to be all 1, based on the equation $G(X) = X^6 + X + 1$.
The 20-bit code is further packaged into 3 bytes according to the following format illustrated in Table 3.

Table 3. Three Byte VARIS Code

	b7	b6	b5	b4	b3	b2	b1	b0
1st Byte	---	---	Word0 B			Word0 A		
2nd Byte	Word2				Word1			
3rd Byte	VID_EN	---	CRC					

The three byte VARIS code is constructed by the ARM as part of the initialization process. The ARM calculates two VARIS codes corresponding to the two possible aspect ratios. The proper code is selected based on the aspect ratio from the bitstream extracted by the video decoder. The user can set VID_EN to signal the NTSC/PAL encoder to enable (1) or disable (0) the VARIS code. The transmission order is the 1st byte first and it is transmitted during the non-active video line and before the transmission of video data.

The timing of the VARIS output is shown in the following Figure 4. The timing of 4:2:2 and 4:4:4 digital video output is shown in Figure 5.

The PCM audio output from the `AV310 is a serial PCM data line, with associated bit and left/right clocks.

PCM data is outputs serially on PCMOUT using the serial clock ASCLK. ASCLK is derived from the PCM clock, PCMCLK, according to the PCM Select bits in the control register. PCM clock must be the proper multiple of the sampling frequency of the bitstream. The PCMCLK may be input to the device or internally derived from an 18.432 MHz clock, depending on the state of the PCM_SRC pin. The data output of PCMOUT alternates between the two channels, as designated by LRCLK as depicted in Figure 6. The data is output most significant bit first. In the case of 18-bit output, the PCM word size is 24 bits. The first six bits are zero, followed by the 18-bit PCM value.

The SPDIF output conforms to a subset of the AES3 standard for serial transmission of digital audio data. The SPDIF format is a subset of the minimum implementation of AES3.

When the PCM_SRC pin is low, the `AV310 generates the necessary output clocks for the audio data, phase locked to the input bitstream. The clock generator requires an 18.432 MHz external VCXO and outputs a control voltage that can be applied to the external loop filter and VCXO to produce the required input. The clock generator derives the correct output clocks, based on the contents of the audio control register bits PCMSEL1-0, as shown in the following table.

Table 4

Audio Clock Frequencies				
PCMSEL1-0	Description	LRCLK (KHz)	ASCLK (MHz)	PCMCLK (MHz)
00	16 bit PCM, no oversampling	48	1.5360	1.5360
01	16 bit PCM, 256 x oversampling	48	1.5360	12.288
10	18 bit PCM, no oversampling	48	2.304	2.304
11	18 bit PCM, 384 x oversampling	48	2.304	18.432

Maximum clock jitter will not exceed 200 ps RMS. An example circuit is shown in Figure 7.

When PCM_SRC is high, the `AV310 expects the correct PCM oversampling clock frequency to be input on PCMCLK.

The SDRAM must be 16-bit wide SDRAM. The `AV310 provides control signals for up to two SDRAMs. Any combination of 4, 8, or 16 Mbit SDRAMs may be used, provided they total at least 16 Mbits. The SDRAM must operate at

an 81 MHz clock frequency and have the same timing parameters as the TI TMS626162, a 16 Mbit SDRAM.

The extension bus interface is a 16-bit bi-directional data bus with a 25-bit address for byte access. It also provides 3 external interrupts, each with it's own acknowledge signal, and a wait line. All the external memories or I/O devices are mapped to the 32-bit address space of the ARM. There are seven internally generated Chip Selects (CSx) for EEPROM memory, DRAM, modem, front panel, front end control, parallel output port, and 1394 Link device. Each CS has its own defined memory space and a programmable wait register which has a default value 1. The number of wait states depends on the content of the register, with a minimum of one wait state. The EXTWAIT signal can also be used to lengthen the access time if a slower device exists in that memory space.

The Extension Bus supports the connection of 7 devices using the pre-defined chip selects. Additional devices may be used by externally decoding the address bus. The following table shows the name of the device, its chip select, address range, and programmable wait state. Every device is required to have tri-stated data outputs within 1 clock cycle following the removal of chip-select.

Table 5

Extension Bus Chip Select			
Chip Select	Byte Address Range	Wait State	Device
CS1	0200 0000 - 03FF FFFF	1 - 5	EEPROM (up to 32 MBytes)
CS2	0400 0000 - 05FF FFFF	N/A	DRAM (up to 32 MBytes)
CS3	0600 0000 - 07FF FFFF	1 - 7	Modem
CS4	0800 0000 - 09FF FFFF	1 - 7	Front Panel
CS5	0A00 0000 - 0BFF FFFF	1 - 7	Front End Device
CS6	0C00 0000 - 0DFF FFFF	1 - 7	1394 Link Device
CS7	0E00 0000 - 0FFF FFFF	1 - 4	Parallel Data Port

CS1 is intended for ARM application code, but writes will not be prevented.

CS2 is read/write accessible by the ARM. It is also accessed by the TC for TPP and bitBLT DMA transfers.

CS3, CS4, CS5, and CS6 all have the same characteristics. The ARM performs reads and writes to these devices through the Extension Bus.

CS7 is read and write accessible by the ARM. It is also accessed by the TC for TPP DMAs, and it is write only. The parallel port is one byte wide and it is accessed via the least significant byte.

The Extension Bus supports connection to external EEPROM, SRAM, or ROM memory and DRAM with its 16-bit data and 25-bit address. It also supports DMA transfers to/from the Extension Bus. DMA transfers within the extension bus are not supported. However, they may be accomplished by DMA to the SRAM, followed by DMA to the extension bus. Extension Bus read and write timing are shown in Figure 8 (read) and Figure 9 (write), both with two programmable wait states. The number of wait state can be a calculated by the following formula:

# of wait states = round_up[((CS_delay + device_cycle_time) /24) - 1]

For example, the CS_delay on the chip is 20 nsec. A device with 80 nsec read timing will need 4 wait states.

There are three interrupt lines and three interrupt acknowledges in the `AV310. These interrupts and interrupts from other modules are handled by a centralized interrupt handler. The interrupt mask and priority are managed by the firmware. The three extension bus interrupts are connected to three different IRQs. When the interrupt handler on the ARM begins servicing one of these IRQs, it should first issue the corresponding EXTACK signal. At the completion of the IRQ, the ARM should reset the EXTACK signal.

The EXTWAIT signal is an alternative way for the ARM to communicate with slower devices. It can be used together with the programmable wait state, but it has to become active before the programmable wait cycle expires. The total amount of wait states should not exceed the maximum allowed from Table 5. If the combined total wait states exceeds its maximum, the decoder his not guaranteed to function properly. When a device needs to use the EXTWAIT signal, it should set the programmable wait state to at least 2. Since the EXTWAIT signal has the potential to stall the whole decoding process, the ARM will cap its waiting to 490 nanoseconds. Afterwards, the ARM assumes the device that generated the EXTWAIT has failed and will ignore EXTWAIT from then on. Only a software or hardware reset can activate the EXTWAIT signal again. The timing diagram of a read with EXTWAIT signal on is shown in the Figure 10.

The Extension Bus supports access to 70ns DRAM with 2 wait states. The DRAM must have a column address that is 8-bit, 9-bit, or 10-bit. The DRAM must have a data width of 8 or 16 bits. Byte access is allowed even when the DRAM

has a 16 bit data width. The system default DRAM configuration is 9-bit column address and 16-bit data width. The firmware will verify the configuration of DRAM during start up.

The 'AV310 includes an Inter Integrated Circuit (I^2C) serial bus interface that can act as either a master (default) or slave. Only the 'standard mode' (100 kbit/s) I^2C-bus system is implemented; 'fast mode' is not supported. The interface uses 7-bit addressing. When in slave mode, the address of the 'AV310 is programmed by the API.

Timing for this interface matches the standard timing definition of the I^2C bus.

The 'AV310 includes two general purpose 2-wire UARTs that are memory mapped and fully accessible by application programs. The UARTs operate ink asynchronous mode only and support baud rates of 1200, 2400, 4800, 9600, 14400, 19200 and 28800 kbps. The outputs of the UARTs are digital and require external level shifters for RS232 compliance.

The IR, RF, and SIRCSI ports require a square wave input with no false transitions; therefore, the signal must be thresholded prior to being applied to the pins. The interface will accept an IR, RF, or SIRCSI data stream up to a frequency of 1.3 KHz. Although more than one may be active at any given time, only one IR, RF, or SIRCSI input will be decoded. Decoding of the IR, RF, and SIRCSI signals will be done by a combination of hardware and software. See the Communications Processor Module for further details.

SIRCSO outputs the SIRCSI or IR input or application-generated SIRCSO codes.

The 'AV310 provides a dedicated data interface for 1394. To complete the implementation, the 'AV310 requires an external packetizer, Link layer, and Physical layer devices. Figure 11 depicts the connection.

The control/command to the packetizer or the Link layer interface device is transmitted via the Extension Bus. The 1394 data is transferred via the 1394 interface which has the following 14 signals:

Table 6

1394 Interface Signals		
Signal Name	I/O	Description
PDATA (8)	I/O	8 bit data
PWRITE (1)	O	if PWRITE is high (active) the 'AV310 writes to the Link device
PPACEN (1)	I/O	asserted at the beginning of a packet and remains asserted during the time of packet transfer
PREADREQ (1)	I	asserted (active high) if the Link device is ready to output to the 'AV310 and the stamped time comes
PREAD (1)	O	if PREAD is high (active) the 'AV310 reads from the Link device
CLK40 (1)	O	40.5 MHz clock. Wait states can be used to slow data transfer.
PERROR (1)	I/O	indicates a packet error

In recording mode, the 'AV310 will send either encrypted or clean packets to the 1394 interface. The packet is transferred as it comes in. When recording encrypted data, the TPP will send each byte directly to the 1394 interface and bypass the DES module. In the case of recording decrypted data, the TPP will send the packet payload to the DES module, then forward a block of packets to the 1394 interface. The interface sends the block of packets out byte by byte. No processing will be done to the packet during recording, except setting the encrypt bit to the proper state. In particular, the TPP will not remove CWP from the Auxiliary packet. During playback mode, the packet coming from the interface will go directly into the TPP module. Figure 12 shows the functional block diagram of the data flow between the TPP, DES, and 1394 interface. The packet coming out from TPP scan go either to the 1394 interface or to the RAM through Traffic Controller, or to both places at the same time. This allows the 'AV310 to decode one program while recording from 1 to all 32 possible services from a transponder.

Figure 13 and Figure 14 depict the read and write timing relationships on the 1394 interface.

During recording, if the DERROR signal from the front end interface goes high in the middle of a packet, it is forwarded to the PERROR pin. If DERROR becomes active in between packets, then a PERROR Signal will be generated during the transfer of the next packet for at least one PDATA cycle.

During playback mode, the external 1394 device can only raise the PERROR signal when the PPACEN is active to indicate either error(s) in the current packet or that there are missing packet(s) prior to the current one. PERROR is ignored unless the PPACEN is active. The PERROR signal should stay high for at least two PCLK cycles. There should be at most one PERROR signal per packet.

The 'AV310 requires a hardware reset on power up. Reset of the device is initiated by pulling the RESET pin low, while the clock is running, for at least 100 ns. The following actions will then occur: input data on all ports will be ignored; external memory is sized; data pointers are reset; all modules are initialized and set to a default state: the TPP tables are initialized; the audio decoder is set for 16 bit output with 256 x oversampling; the OSD background color is set to blue and video data is selected for both the analog and digital outputs; MacroVision is disabled; and the I^2C port is set to master mode.

When the reset sequence is finished, the device will begin to accept data. All data input prior to the end of the reset sequence will be ignored.

JTAG boundary scan is included in the 'AV310. Five pins (including a test reset) are used to implement the IEEE 1149.1 (JTAG) specification. The port includes an 8-bit instruction register used to select the instruction. This register is loaded serially via the TDI input. Four instructions are supported, all others are ignored: Bypass; Extest; Intest and Sample.

Timing for this interface conforms to the IEEE 1149.1 specification.

Features of the ARM/CPU module: runs at 40.5 MHz; Supports byte (8-bit), half-word (16-bit), and word (32-bit) data types; reads instructions from on-chip ROM or from the Extension Bus; can switch between ARM (32-bit) or Thumb (16-bit) instruction mode; 32-bit data and 32-bit address lines; 7 processing modes; and two interrupts, FIQ and IRQ.

The CPU in the 'AV310 is a 32 bit RISC processor, the ARM7TDMI/ Thumb, which has the capability to execute instructions in 16 or 32 bit format at a clock frequency of 40.5 MHz. The regular ARM instructions are exactly one word (32-bit) long, band the data operations are only performed on word quantities. However, LOAD and STORE instructions can transfer either byte or word quantities.

The Thumb uses the same 32 bit architecture with an 16-bit instruction set. That is, it retains the 32-bit performance but reduces the code size with 16-bit instructions. With 16-bit instruction, Thumb still gives 70 - 80% of the performance of the ARM when running ARM instructions from 32-bit memory. In this document, ARM and Thumb are used interchangeably.

ARM uses a LOAD and STORE architecture, i.e. all operations are on the registers. ARM has 6 different processing modes, with 16 32-bit registers visible in user mode. In the Thumb state, there are only 8 registers available in user mode. However, the high registers may be accessed through special instructions. The instruction pipeline is three stage, fetch → decode → execute, and most instructions only take one cycle to execute. Figure 15 shows the data path of ARM processor core.

The ARM CPU is responsible for managing all the hardware and software resources in the 'AV310. At power up the ARM will verify the size of external memory. Following that, it will initialize all the hardware modules by setting up control registers, tables, and reset data pointers. It then executes the default firmware from internal ROM. A set of run-time library routines provides the access to the firmware and hardware for user application programs. The application programs are stored in external memory attached to the Extension Bus.

During normal operation the ARM constantly responds, based on a programmable priority, to interrupt requests from any of the hardware modules and devices on the Extension Bus. The kind of interrupt services include transport packet parsing, program clock recovery, traffic controller and OSD service requests, service or data transfer requests from the Extension Bus and Communication Processor, and service requests from the Audio/Video decoder.

Features of the Traffic Controller Module: manages interrupt requests; authorizes and manages DMA transfers; provides SDRAM interface; manages Extension Bus; provides memory access protection; manages the data flow between processors and memories: TPP/DES to / from internal Data RAM; Data RAM to/from Extension Bus; SDRAM to OSD; OSD to/from Data RAM; Audio/ Video Decoder to/from SDRAM; and SDRAM to/from Data RAM. Generates chip selects (CS) for all internal modules and devices on the Extension Bus; generates programmable wait states for devices on the Extension Bus; and provides 3 breakpoint registers and 64 32-bit patch RAM.

Figure 16 depicts the data flow managed by the Traffic Controller.

The SDRAM interface supports 12 nanoseconds 16-bit data width SDRAM. It has two chip selects that allow connections to a maximum of two SDRAM chips. The minimum SDRAM size required by the decoder is 16 Mbit. Other supported sizes and configurations are:

16 Mbit → one 16 Mbit SDRAM
20 Mbit → one 16 Mbit and one 4 Mbit SDRAM
24 Mbit → one 16 Mbit and one 8 Mbit SDRAM
32 Mbit → two 16 Mbit SDRAM

The access to the SDRAM can be by byte, half word, single word, continuous block, video line block, or 2D macroblock. The interface also supports decrement mode for bitBLT block transfer.

The two chip selects correspond to the following address ranges:

SCS1 → 0xFE00 0000 - 0xFE1F FFFF
SCS2 → 0xFE20 0000 - 0xFE3F FFFF

During decoding, the 'AV310 allocates the 16 Mbit SDRAM for NTSC mode according to Table 7.

Table 7

Memory Allocation of 16 Mbit SDRAM (NTSC)		
Starting Byte Address	Ending Byte Address	Usage
0x000000	0x0003FF	Pointers
0x000400	0x000FFF	Tables and FIFOs
0x001000	0x009FFF	Video Microcode (36,864 bytes)
0x00A000	0x0628FF	Video Buffer (2,902,008 bits)*
0x062900	0x0648FF	Audio Buffer (65,536 bits)
0x064900	0x0E31FF	First Reference Frame (518,400 bytes)
0x0E3200	0x161CFF	Second Reference Frame (518,400 bytes)
0x161D00	0x1C9DFF	B Frame (426,240 bytes, 0.82 frames)
0x1C9E00	0x1FFFFF	OSD or other use (222,210 bytes)*

* These values are for the current DSS specification. In the latest proposed specification, the VBV buffer size is reduced to 1,835,008 bits, giving 355,586 bytes for OSD or other use.

However, it is also within the sccope of the present invention to put the VBV buffer in optional memory on the extension bus 300 and thereby free up the SDRAM memory by the amount of the VBV buffer. This means that the SDRAM is allocated in a different manner than that of Table 7; that is the OSD memory size may be expanded or any of the other blocks expanded.

Interrupt requests are generated from internal modules like the TPP, OSD, A/V decoder and Communication Processor, and devices on the Extension Bus. Some of the requests are for data transfers to internal RAM, while others are true interrupts to the ARM CPU. The Traffic Controller handles data transfers, and the ARM provides services to true interrupts. The interrupts are grouped into FIQ and IRQ. The system software will use FIQ, while the application software will use IRQ. The priorities for FIQs and IRQs are managed by the firmware.

The SDRAM is used to store system level tables, video and audio bitstreams, reconstructed video images, OSD data, and video decoding codes, tables, and FIFOs. The internal Data RAM stores temporary buffers, OSD window attributes, keys for conditional access, and other tables and buffers for firmware. The TC manages two physical DMA channels, but only one of them, the General Purpose DMA, is visible to the user. The user has no knowledge of the DMAs initiated by the TPP, the video and audio decoder, and the OSD module. The General Purpose DMA includes ARM-generated and bitBLT-generated DMAs. The TC can accept up to 4 general DMAs at any given time. Table 8 describes the allowable General Purpose DMA transfers.

Table 8

DMA Sources and Destinations DMA Transfer			
	SDRAM	Data RAM	Extension Bus
SDRAM	NO	YES	NO
Data RAM	YES	NO	YES
Extension Bus	NO	YES	NO

Note that there is no direct DMA transfer to/from the Extension Bus memories from/to the SDRAM. However, the user can use the bitBLT hardware which uses Data RAM as intermediate step for this purpose. The only constraint is the block being transferred has to start at a 32-bit word boundary.

Features of the TPP Module: parses transport bitstreams; accepts bitstream either from the front end device or from the 1394 interface; performs System Clock Reference (SCR) recovery; supports transport stream up to 40 Mbits-

per-second; accepts 8-bit parallel input data; supports storage of 32 SCID; lost-packet detection; provides decrypted or encrypted packets directly to the 1394 interface; and internal descrambler for DSS with the Data Encryption Standard (DES) implemented in hardware.

The TPP accepts packets byte by byte. Each packet contains a unique ID, SCID, and the TPP extracts those packets containing thee designated ID numbers. It processes the headers of transport packets and transfers the payload or auxiliary packets to the internal RAM via the DES hardware and Traffic Controller. Special firmware running on the ARM handles DES key extraction and activates DES operation. The ARM/CPU performs further parsing on auxiliary packets stored in the internal RAM. The ARM and TPP together also perform SCR clock recovery. Figure 17 is an example circuit for the external VCXO. The output from the 'AV310 is a digital pulse with 256 levels.

The Conditional Access and DES block is part of the packet header parsing function. A CF bit in the header indicates whether the packet is clean or has been encrypted. The clean packet can be forwarded to the internal RAM directly, while the encrypted one needs to go through the DES block for decryption. The authorization and decryption key information are transmitted via Control Word Packet (CWP). An external Smart Card guards this information and provides the proper key for the DES to work.

The 1394 interface is directly connected to the TPP/DES module. At the command of the user program, the TPP/DES can send either clean or encrypted packets to the 1394 interface. The user can select up to 32 services to record. If the material is encrypted, the user also needs to specify whether to record clean or encrypted video. In recording mode, the TPP will appropriately modify the packet header if decrypted mode is selected; in encrypted mode, the packet headers will not be modified. During the playback mode, the 1394 interface forwards each byte as it comes in to the TPP. The TPP parses the bitstream the same way it does data from the front end.

Features of Video Decoder Module: Real-time video decoding of MPEG-2 Main Profile Main level and MPEG-1; error detection and concealment; internal 90 KHz/27 MHz System Time Clock; sustained input rate of 16 Mbps; supports Trick Mode with full trick mode picture; provides 1/4 and 1/16 decimated size picture; extracts Closed Caption and other picture user data from the bitstream; 3:2 pulldown in NTSC mode; and supports the following display format with polyphase horizontal resampling and vertical chrominance filtering

Table 9

Supported Video Resolutions			
NTSC (30 Hz)		PAL (25 HZ)	
Source	Display	Source	Display
720 x 480	720 x 480	720 x 576	720 x 576
704 x 480	720 x 480	704 x 576	720 x 576
544 x 480	720 x 480	544 x 576	720 x 576
480 x 480	720 x 480	480 x 576	720 x 576
352 x 480	720 x 480	352 x 576	720 x 576
352 x 240	720 x 480	352 x 288	720 x 576

Pan-and-scan for 16:9 source material according to both DSS and MPEG syntax; high level command interface; and synchronization using Presentation Time Stamps (PTS).

The Video Decoder module receives a video bitstream from SDRAM. It also uses SDRAM as its working memory to store tables, buffers, and reconstructed images. The decoding process is controlled by a RISC engine which accepts high level commands from the ARM. In that fashion, the ARM is acting as an external host to initialize and control Video Decoder module. The output video is sent to the OSD module for further blending with OSD data.

Besides normal bitstream decoding, the Video decoder also extracts from the picture layer user data the Closed Caption (CC), the Extended Data Services (EDS), the Presentation Time Stamps (PTS) and Decode Time Stamps, the pan_and_scan, the fields display flags, and the no_burst flag. These data fields are specified by the DSS. The CC and EDS are forwarded to the NTSC/PAL encoder module and the PTS is used for presentation synchronization. The other data fields form a DSS-specific constraints on the normal MPEG bitstream, and they are used to update information obtained from the bitstream.

When the PTS and SCR (System Clock Reference) do not match within tolerance, the Video decoder will either redisplay or skip a frame. At that time, the CC/EDS will be handled as follows: if redisplaying a frame, the second display will not contain CC/EDS; if skipping a frame, the corresponding CC/EDS will also be skipped. During trick mode decod-

ing, the video decoder repeats the following steps: searches for a sequence header followed by an I picture; ignores the video buffer underflow error; and continuously displays the decoded I frame.

Note that trick mode I frame data has to contain the whole frame instead of only several intra slices.

The Video decoder accepts the high level commands detailed in Table 10.

Table 10

Video Decoder Commands	
Play	normal decoding
Freeze	normal decoding but continue to display the last picture
Stop	stops the decoding process. The display continue with the last picture
Scan	searches for the first I picture, decodes it, continuously displays it, and flushes the buffer
NewChannel	for channel change. This command should be preceded by a Stop command.
Reset	halts execution of the current command. The bitstream buffer is flushed and the video decoder performs an internal reset
Decimatel/2	continue normal decoding and displaying of a 1/2 x 1/2 decimated picture (used by OSD API)
Decimatel/4	continue normal decoding and displaying of a 1/4 x 1/4 decimated picture (used by OSD API)

The following table shows the supported aspect ratio conversions.

Table 11

Aspect Ratio Conversions		
Source	Display	
	16:9	4:3
4:3	YES	NO
16:9	PAN-SCAN	YES

The Pan-Scan method is applied when displaying a 16:9 source video on a 4:3 device. The Pan-Scan location specifies to the 1, 1/2, or 1/4 sample if the source video has the full size, 720/704 x 480. If the sample size is smaller than full then the Pan-Scan location only specifies to the exact integer sample. Note that the default display format output from 'AV310 is 4:3. Outputting 16:9 video is only available when the image size is 720/704 x 480. A reset is also required when switching between a 4:3 display device and a 16:9 one.

The 1/2 and 1/4 decimation, in each dimension, is supported for various size images in 4:3 or 16:9 format. The following table provides the details.

Table 12

Decimation Modes						
Size Size	Source					
	4:3			16:9		
	Full	1/2	1/4	Full	1/2	1/4
720/704 x 480	YES	YES	YES	YES	YES	YES
544 x 480	YES	YES	YES	YES	YES	YES
480 x 480	YES	YES	YES	YES	YES	YES
352 x 480	YES	YES	YES	YES	YES	YES

Table 12 (continued)

Decimation Modes						
Size Size	Source					
	4:3			16:9		
	Full	1/2	1/4	Full	1/2	1/4
352 x 240	YES	YES	YES	NO	NO	NO

Features of the audio decoder module: decodes MPEG audio layers 1 and 2; supports all MPEG-1 and MPEG-2 data rates and sampling frequencies, except half frequency; provides automatic audio synchronization; supports 16- and 18-bit PCM data; outputs in both PCM and SPDIF formats; generates the PCM clock or accepts an external source; provides error concealment (by muting) for synchronization or bit errors; and provides frame-by-frame status information.

The audio module receives MPEG compressed audio data from the traffic controller, decodes it, and outputs audio samples in PCM format. The ARM CPU initializes/controls the audio decoder via a control register and can read status information from the decoder's status register.

Audio frame data and PTS information is stored in the SDRAM in packet form. The audio module will decode the packet to extract the PTS and audio data.

The ARM can control the operation of the audio module via a 32-bit control register. The ARM may reset or mute the audio decoder, select the output precision and oversampling ratio, and choose the output format for dual channel mode. The ARM will also be able to read status information from the audio module. One (32-bit) register provides the MPEG header information and sync, CRC, and PCM status.

The audio module hash two registers: a read/write control register and a read-only status register. The registers are defined below.

Table 13. Audio Module Registers

Register #	Location	Description
0 (Control Register - R/W)	31:6	Reserved (set to 0)
	5:4	PCM Select 00 = 16 bit, no oversampling 01 = 16 bit, 256 x oversampling 10 = 18 bits, no oversampling 11 = 18 bits, 384 x oversampling
	3:2	Dual Channel Mode Output Mode Select 00 = Ch 0 on left, Ch 1 on right 01 = Ch 0 on both left and right 10 = Ch 1 on both left and right 11 = Reserved
	1	Mute 0 = Normal operation 1 = Mute audio output
	0	Reset 0 = Normal operation 1 = Reset audio module
1 (Status Register - R only)	31	Stereo Mode 0 = all other 1 = dual mode
	30:29	Sampling Frequency 00 = 44.1 KHz 01 = 48 KHz 10 = 32 KHz 11 = Reserved
	28:27	De-emphasis Mode 00 = None 01 = 50/15 microseconds 10 = Reserved 11 = CCITT J.17
	26	Synchronization Mode

Register #	Location	Description
		0 = Normal operation 1 = Sync recovery mode
	25	CRC Error 0 = No CRC error or CRC not enabled in bitstream 1 = CRC error found
	24	PCM Underflow 0 = Normal operation 1 = PCM output underflowed
	23:4	Bits 19-0 of the MPEG header
	3:0	Version number of the audio decoder

Features of the OSD module: supports up to 8 hardware windows, one of which can be used for a cursor; all the non-overlapped windows can be displayed simultaneously; overlapped windows are displayed obstructively with the highest priority window on top; provides a hardware window-based rectangle cursor with programmable size and blinking frequency; and provides a programmable background color, which defaults to blue; supports 4 window formats (empty window for decimated video; bitmap; YCrCb 4:4:4 graphics component; and YCrCb 4:2:2 CCIR 601 component); supports blending of bitmap, YCrCb 4:4:4, or YCrCb 4:2:2 with motion video and with an empty window; supports window mode and color mode blending; provides a programmable 256 entries Color Look Up table; outputs motion video or mixture with OSD in a programmable 422 or 444 digital component format; provides motion video or mixture with OSD to the on-chip NTSC/PAL encoder and provides graphics acceleration capability with bitBLT hardware Each hardware window has the following attributes: window position (any even pixel horizontal position on screen; windows with decimated video have to start from an even numbered video line also); window size: from 2 to 720 pixel wide (even values only) and 1 to 576 lines; window base address; data format (bitmap, YCrCb 4:4:4, YCrCb 4:2:2, and empty); bitmap resolution (1, 2, 4, and 8 bits per pixel); full or half resolution for bitmap and YCrCb 4:4:4 windows; bitmap color palette base address; blend enable flag; 4 or 16 levels of blending; transparency enable flag for YCrCb 4:4:4 and YCrCb 4:2:2; and output channel control.

The OSD module is responsible for managing OSD data from different OSD windows and blending them with the video. It accepts video from the Video Decoder, reads OSD data from SDRAM, and produces one set of video output to the on-chip NTSC/PAL Encoder and another set to the digital output that goes off the chip. The OSD module defaults to standby mode, in which it simply sends video from the Video Decoder to both outputs. After being activated by the ARM CPU, the OSD module, following the window attributes set up by the ARM, reads OSD data and mixes it with the video output. The ARM CPU is responsible for turning on and off OSD operations. The bitBLT hardware which is attached to the OSD module provides acceleration to memory block moves and graphics operations. Figure 18 shows the block diagram of the OSD module. The various functions of the OSD are described in the following subsections.

The OSD data has variable size. In the bitmap mode, each pixel can be 1, 2, 4, or 8 bits wide. In the graphics YCrCb 4:4:4 or CCIR 601 YCrCb 4:2:2 modes, it takes 8-bit per components, and the components are arranged according to 4:4:4 (Cb/Y/Cr/Cb/Y/Cr) or 4:2:2 (Cb/Y/Cr/Y) format. In the case where RGB graphics data needs to be used as OSD, the application should perform software conversion to Y/Cr/Cb before storing it. The OSD data is always packed into 32-bit words and left justified. Starting from the upper left corner of the OSD window, all data will be packed into adjacent 32-bit words. The dedicated bitBLT hardware expedites the packing and unpacking of OSD data for the ARM to access individual pixels, and the OSD module has an internal shifter that provides pixel access.

In NTSC mode, the available SDRAM is able to store one of the following OSD windows with the size listed in Table 14, with the current and proposed VBV buffer size for DSS.

Table 14

SDRAM OSD Window Size		
bits/pixel	720 x 480 frames	
	Current	Proposed
24	0.21	0.34
8	0.64	1.03
4	1.29	2.06
2	2.58	4.12

An OSD window is defined by its attributes. Besides storing OSD data for a window into SDRAM, the application program also needs to update window attributes and other setup in the OSD module as described in the following subsections.

The CAM memory contains X and Y locations of the upper left and lower right corners of each window. The application program needs to set up the CAM and enable selected OSD windows. The priority of each window is determined by its location in the CAM. That is, the lower address window always has higher priority. In order to swap the priority of windows, the ARM has to exchange the locations within the CAM.

The OSD module keeps a local copy of window attributes. These attributes allow the OSD module to calculate the address for the OSD data, extract pixels of the proper size, control the blending factor, and select the output channel.

Before using bitmap OSD the application program has to initialize the 256 entry color look up table (CLUT). . The CLUT is mainly used to convert bitmap data into Y/Cr/Cb components. Since bitmap pixels can have either 1, 2, 4, or 8 bits, the whole CLUT can also be programmed to contain segments of smaller size tables, such as sixteen separate, 16-entry CLUTs.

There are two blending modes. The window mode blending applies to OSD window of type bitmap, YCrCb 4:4:4, and YCrCb 4:2:2. The color mode, pixel by pixel, blending is only allowed for the bitmap OSD. Blending always blends OSD windows with real time motion video. That is, there is no blending among OSD windows except the empty window that contains decimated motion video. In case of overlapping OSD windows the blending only occurs between the top OSD window and the video. The blending is controlled by the window attributes, Blend_En (2-bit), Blend Level (4-bit), and Trans_En (1-bit). Blend_En activates blending as shown in Table 15. In window mode all pixels are mixed with the video data based on the level defined by the attributes Blend Level. In the color mode the blending level is provided in the CLUT. That is, the least significant bit of Cb and Cr provides the 4 level blending, while the last two bits from Cb and Cr provide the 16 level blending. Transparency level, no OSD but only video, is achieved with the Trans_En bit on and the OSD pixel containing all 0s.

Table 15

OSD Blending Control	
Blend_En	Blending modes
00	Disable Blending
01	4 Level Color Blending
10	16 Level Color Blending
11	Window Mode Blending

A rectangular blinking cursor is provided using hardware window 0. With window 0, the cursor always appears on top of other OSD Windows. The user can specify the size of the cursor via window attribute. The activation of the cursor, its color, and blinking frequency are programmable via control registers. When hardware window 0 is designated as the cursor, only seven windows are available for the application. If a hardware cursor is not used, then the application can use window 0 as a regular hardware window.

After the OSD windows are activated, each of them has an attribute, Disp_Ch_Cntl[1,0], that defines the contents of the two output channels (the analog and digital video outputs) when the position of that window is currently being dis-

played. The following table shows how to control output channels.

Table 16

OSD Module Output Channel Control			
Disp_Ch_c ntl[1]	Disp_Ch_cnt l[0]	Channel 1 Digital Video Output	Channel 0 To NTSC/PAL Encoder
0	0	MPEG Video	MPEG Video
0	1	MPEG Video	Mixed OSD_Window
1	0	Mixed OSD_Window	MPEG Video
1	1	Mixed OSD_Window	Mixed OSD_Window

Example displays of these two output channels are shown in Figure 19.

The bitBLT hardware provides a faster way to move a block of memory from one space to the other. It reads data from a source location, performs shift/mask/merge/expand operations on the data, and finally writes it to a destination location. This hardware enables the following graphics functions: Set/Get Pixel; Horizontal/Vertical Line Drawing; Block Fill; Font BitBLTing; Bitmap/graphic BitBLTing; and Transparency.

The allowable source and destination memories for bitBLT are defined in Table 17.

Table 17

Source and Destination Memories for BitBLT		
Source Memory	Destination Memory	
	SDRAM	Ext_Bus Memory
SDRAM	YES	YES
Ext_Bus Memory	YES	YES

The types of source and destination OSD windows supported by the bitBLT are the given in the following table (the HR stands for half resolution).

Table 18

Allowable BitBLT Window Formats					
Source OSD Window	Destination OSD Window				
	YCrCb 4:4:4	YCrCb 4:4:4_HR	YCrCb 4:2:2	Bitmap	Bitmap_H R
YCrCb 4:4:4	YES	YES	NO	NO	NO
YCrCb 4:4:4_HR	YES	YES	NO	NO	NO
YCrCb 4:2:2	NO	NO	YES	NO	NO
Bitmap	YES	YES	NO	YES	YES
Bitmap_HR	YES	YES	NO	YES	YES

Since the bitmap allows resolutions of 1, 2, 4, or 8 bits per pixel, the bitBLT will drop the MSB bits or pad it with 0s when swapping between windows of different resolution. For half-resolution OSD, the horizontal pixel dimension must be even numbers. For YCrCb 4:2:2 data, the drawing operation is always on 32-bit words, two adjacent pixels that align with the word boundary.

In a block move operation, the block of data may also be transparent to allow text or graphic overlay. The pixels of the source data will be combined with the pixels of the destination data. When transparency is turned on and the value of the source pixel is non-zero, the pixel will be written to the destination. When the value of the pixel is zero, the desti-

nation pixel will remain unchanged. Transparency is only allowed from bitmap to bitmap, and from bitmap to YCrCb 4:4:4.

Features of NTSC/PAL Encoder module: supports NTSC and PAL B, D, G/H, and I display formats; outputs Y, C, and Composite video with 9-bit DACs; complies to the RS170A standard; supports MacroVision Anti-taping function; provides Closed Caption, Extended Data Services, and aspect ratio VARIS encoding; and provides sync signals with option to accept external sync signals.

This module accepts from the OSD module the video data that may have been blended with OSD data and converts it to Y, C, and Composite analog outputs. The Closed Caption and Extended Data Services data are provided by the Video decoder through a serial interface line. These data are latched into corresponding registers. The CC encoder sends out Closed Caption data at video line 21 and Extended Data Services at video line 284. The ARM initializes and controls this module via the ARM Interface block. It also sends VARIS code to the designated registers which is then being encoded into video line 20. The ARM also turns on and off MacroVision through the ARM Interface block. The default state of MacroVision is off.

Features of the Communications Processor module; provides two programmable timers; provides 3 UARTs - one for Smart Card and two for general use; accepts IR, SIRCSI and RF signals; provides a SIRCSO output; provides two general purpose I/Os; and manages I^2C and JTAG interfaces.

This module contains a collection of buffers, control registers, and control logic for various interfaces, such as UARTs, IR/RF, I^2C, and JTAG. All the buffers and registers are memory mapped and individually managed by the ARM CPU. Interrupts are used to communicate between these interface modules and the ARM CPU.

The `AV310 has two general purpose timers which are user programmable. Both timers contain 16 bit counters with 16 bit pre-scalers, allowing for timing intervals of 25 ns to 106 seconds. Each timer, timer0 and timer1, has an associated set of control and status registers. These registers are defined in Table 19.

Table 19. Timer Control and Status Registers

Register Name	Read/Write	Description
tcrx	R/W	Timer Control Register
		31 - Reserved (set to 0) 6 tint_mask 5 0 = enable interrupts 1 = mask interrupts reserved (set to 1) 4 reserved 3 soft - soft stop: 2 0 = reload counters on 0 1 = stop timer on 0 1 tss - timer stop: 0 = start 1 = stop 0 trb - timer reload 0 = do not reload 1 = reload the timer (read 0)
tddrx	W	Timer Divide Down (15-0). Contains the value for the pre-scalar to preload psc during pre-scalar rollover. (Note: reading this register is equivalent to reading the prld register.)
prdx	W	Timer Period Register (15-0). Contains the value for tim to preload during tim rollover. (Note: reading this register is equivalent to reading the tim32 register.)
preldx	R	Preload Value.
		31 - Value of prd 16 Value of tddr 16 -

Register Name	Read/Write	Description 0
tim32x	R	Actual Time Value (31-0)
		31 - Value of tim 16 Value of psc 16 - 0

Note: x designates the timer number, 0 or 1.

The timers are count-down timers composed of 2 counters: the timer pre-scaler, psc, which is pre-loaded from tddr and counts down every sys_clock ; and the timer counter, tim, (pre-loaded from prd). When psc = 0, it pre-loads itself and decrements tin by one. This divides the sys_clock by the following values:

(tddr + 1) * (prd + 1),if tddr and prd are not both 0, or 2, if tddr and prd are both 0.

When tim = 0 and psc = 0, the timer will issue an interrupt if the corresponding tint_mask is not set. Then both counters are pre-loaded if soft = 0. If soft is 1, the timer stops counting.

The timer control register (tcr) can override normal timer operations. The timer reload bit, trb, causes both counters to pre-load, while the timer stop bit, tss, causes both counters to stop.

The two general purpose 2-wire UARTs are asynchronous mode, full duplex, double buffered with 8 bytes FIFO UARTs that operate at up to 28.8 kbps. They transmit/receive 1 start bit, 7 or 8 data bits, optional parity, and 1 or 2 stop bits.

The UARTs are fully accessible to the API and can generate interrupts when data is received or the transmit buffer is empty. The ARM also has access to a status register for each UART that contains flags for such errors as data overrun and framing errors.

The IR/RF remote control interface is a means of transmitting user commands to the set top box. This interface consists of a custom hardware receiver implementing a bit frame-based communication protocol. A single bit frame represents a user command.

The bit frame is defined in three possible lengths of 12, 15 or 20 bits. The "ON/OFF" values of the bits in the frame are represented by two different length pulse widths. A 'one' is represented by a pulse width of 1.2 ms and a 'zero' is represented by a 0.6 ms pulse width. The example in Figure 20 shows the IR input bitstream. The bitstream is assumed to be free of any carrier (36-48 KHz typical) and represents a purely digital bitstream in return-to-zero format. The hardware portion of this interface is responsible for determining the bit value along with capturing the bit stream and placing the captured value into a read register for the software interface to access. Each value placed in the read register will generate an interrupt request.

Each user command is transmitted as a single bit frame and each frame is transmitted a minimum of three times. The hardware interface is responsible for recognizing frames and filtering out unwanted frames. For a bit frame to be recognized by the hardware interface it must pass the following steps: first it must match the expected frame size, 12, 15 or 20 bits; then two of the minimum three frames received must match in value. A frame match when detected by the hardware interface will generate only one interrupt request.

The IR/RF protocol has one receive interrupt, but it is generated to indicate two different conditions. The two different conditions are start and finish of a user command. The first type of receive interrupt (start) is generated when the hardware interface detects a new frame (remember 2 out of three frames must match). The second type of interrupt is generated when there has been no signal detected for the length of a hardware time out period (user command time out). Each frame, when transmitted, is considered to be continuous or repeated. So although there is a three frame minimum for a user command the protocol is that when a start interrupt is received the interface will assume that until a finish (time out) interrupt is generated the same frame is being received.

A typical example of the receive sequence is to assume that the interface has been dormant and the hardware interface detects a signal that is recognized as a frame. This is considered the start of a user command, and a start interrupt is issued by the hardware interface. The finish of a user command is considered to be when there has not been

a signal detected by the hardware interface for a time out period of approximately 100 ms. The finish will be indicated by an interrupt from the hardware interface.

During a receive sequence it is possible to receive several start interrupts before receiving a finish interrupt. Several start interrupts maybe caused by the user entering several commands before the time out period has expired. Each of these commands entered by the user would be a different command. A new user command can be accepted before the previous command time out .

The IR, SIRCSI, and RF inputs share common decoding logic. Figure 21 shows a theoretical model of the hardware interface. There are three possible inputs, SIRCSI, IR and RF, and one output, SIRCSO. The IR receiver receives its input from the remote control transmitter while the SIRCSI receives its input from another device's SIRCSO. Again, examining Figure 21 shows that normal operation will have the IR connected to the SIRCSO and the decoder. The SIRCSI signal has priority over the IR and will override any IR signal in progress. If a SIRCSI signal is detected, the hardware interface will switch the input stream from IR to SIRCSI and the SIRCSI will be routed to the decoder and the SIRCSO.

There are two possible inputs for the IR frame type and one input for the RF frame type. A selection must be made by the user if the received frame type is going to be IR or RF. The IR/RF interface contains two 32-bit data registers, one for received data (IRRF Data Decode register) and one for data to be written out (IRRF Encode Data register). In both registers, bits 31-20 are not used and are set to 0.

The `AV310 has two general purpose I/O pins (IO1 and IO2) which are user configurable. Each I/O port has its own 32-bit control/status register, iocsr1 or iocsr2.

If an I/O is configured as an input and the delta interrupt mask is cleared, an ARM interrupt is generated whenever an input changes state. If the delta interrupt mask is set, interrupts to the ARM are disabled. If no other device drives the I/O pin while it is configured as an input, it will be held high by an internal pull-up resistor.

If an I/O is configured as an output (by setting the cio bit in the corresponding control/status register), the value contained in the io_out bit of the control/status register is output. Interrupt generation is disabled when an I/O is configured as an output.

The definition of the control/status registers is given in Table 20.

Table 20. I/O Control/Status Registers

Bit Number	Name	Description
31-4	Reserved	Set to 0 (read only)
3	io_in	input sample value (read only)
2	dim	delta interrupt mask: 0 = generate interrupts 1 = mask interrupts
1	cio	configure i/o: 0 = input 1 = output
0	io_out	output value if cio is 1

The `AV310 includes an $I^2C$ serial bus interface that can act as either a master or slave. (Master mode is the default). In master mode, the `AV310 initiates and terminates transfers and generates clock signals.

To put the device in slave mode, the ARM must write to a control register in the block. The API must set the slave mode select and a 7-bit address for the `AV310. It must also send a software reset to the I2C to complete the transition to slave mode.

In slave mode, when the programmable address bits match the applied address, the `AV310 will respond accordingly. The `AV310 will also respond to general call commands issued to address 0 (the general call address) that change the programmable part of the slave address. These commands are 0x04 and 0x06. No other general call commands will be acknowledged, and no action will be taken.

The circuitry is presently preferably packaged in a 240 pin PQFP. Table 21 is a list of pin signal names and their descriptions. Other pin outs may be employed to simplify the design of emulation, simulation, and/or software debugging platforms employing this circuitry.

Table 21.

Signal Name	#	I/O	Description
Transport Parser			
DATAIN[7:0]*	8	I	Data Input. Bit 7 is the first bit in the transport stream
DCLK*	1	I	Data Clock. The maximum frequency is 7.5 MHz.
PACCLK*	1	I	Packet Clock. Indicates valid packet data on DATAIN.
BYTE_STRT*	1	I	Byte Start. Indicates the first byte of a transport packet for DVB. Tied low for DSS.
DERROR*	1	I	Data Error, active high. Indicates an error in the input data. Tie low if not used.
CLK27*	1	I	27 MHz Clock input from an external VCXO.
VCXO_CTRL*	1	O	VCXO Control. Digital pulse output for external VCXO.
CLK_SEL	1	I	Clock select. CLK_SEL low selects a 27 MHz input clock. When high, selects an 81 MHz input clock.
Communications Processor			
IR*	1	I	Infra-Red sensor input
RF*	1	I	RF sensor input
SIRCSI*	1	I	SIRCS control input

Signal Name	#	I/O	Description
SIRCSO*	1	O	SIRCS control output
UARTDI1*	1	I	UART Data Input, port 1
UARTDO1*	1	O	UART Data Output, port 1
UARTDI2*	1	I	UART Data Input, port 2
UARTDO2*	1	O	UART Data Output, port 2
PDATA	8	I/O	1394 Interface Data Bus
PWRITE	1	O	1394 Interface Write Signal
PREAD	1	O	1394 Interface Read Signal
PPACEN	1	I/O	1394 Interface Packet Data Enable
PREADREQ	1	I	1394 Interface Read Data Request
PERROR	1	I/O	1394 Interface Error Flag
IIC_SDA*	1	I/O	I^2C Interface Serial Data
IIC_SCL*	1	I/O	I^2C Interface Serial Clock
IO1*	1	I/O	General Purpose I/O
IO2*	1	I/O	General Purpose I/O
Extension Bus			
EXTR/W	1	O	Extension Bus Read/Write. Selects read when high, write when low.
EXTWAIT	1	I	Extension Bus Wait Request, active low, open drain
EXTADDR[24:0]	25	O	Extension Address bus: byte address
EXTDATA[15:0]	1	I/O	Extension Data bus

Signal Name	#	I/O	Description
	6	O	
EXTINT[2:0]	3	I	External Interrupt requests (three)
EXTACK[2:0]	3	O	External Interrupt acknowledges (three)
CLK40	1	O	40.5 MHz Clock output for extension bus and 1394 interface
CS1	1	O	Chip Select 1. Selects EEPROM, 32M byte maximum size.
CS2	1	O	Chip Select 2. Selects external DRAM.
CS3	1	O	Chip Select 3. Selects the modem.
CS4	1	O	Chip Select 4. Selects the front panel.
CS5	1	O	Chip Select 5. Selects front end control.
CS6	1	O	Chip Select 6. Selects the 1394 interface.
CS7	1	O	Chip Select 7. Selects the parallel data port.
RAS	1	O	DRAM Row Address Strobe
UCAS	1	O	DRAM Column address strobe for upper byte
LCAS	1	O	DRAM Column address strobe for lower byte
SMIO	1	I/O	Smart Card Input/Output
SMCLK	1	O	Smart Card Output Clock
SMCLK2	1	I	Smart Card Input Clock, 36.8 MHz

Signal Name	#	I/O	Description
SMDETECT	1	I	Smart Card Detect, active low
SMRST	1	O	Smart Card Reset
SMVPPEN	1	O	Smart Card Vpp enable
SMVCCDETECT*	1	I	Smart Card Vcc detect. Signals whether the Smart Card Vcc is on.
SMVCCEN	1	O	Smart Card Vcc enable
Audio Interface			
AUD_PLLI*	1	I	Input Clock for Audio PLL
AUD_PLLO	1	O	Control Voltage for external filter of Audio PLL
PCM_SRC	1	I	PCM Clock Source Select. Indicates whether the PCM clock is input to or generated by the 'AV310.
PCMDATA*	1	O	PCM Data audio output.
LRCLK*	1	O	Left/Right Clock for output PCM audio data.
PCMCLK*	1	I or O	PCM Clock.
ASCLK*	1	O	Audio Serial Data Clock
SPDIF*	1	O	SPDIF audio output
Digital Video Interface			
YCOUT[7:0]	8	O	4:2:2 or 4:4:4 digital video output
YCCLK	1	O	27 or 40.5 MHz digital video output clock
YCCTRL[1:0]	2	O	Digital video output control signal

Signal Name	#	I/O	Description
NTSC/PAL Encoder Interface			
NTSC/PAL	1	I	NTSC/PAL select. Selects NTSC output when high, PAL output when low.
SYNCSEL	1	I	Sync signal select. When low, selects internal sync generation. When high, VSYNC and HSYNC are inputs.
VSYNC	1	I or O	Vertical synchronization signal
HSYNC	1	I or O	Horizontal synchronization signal
YOUT	1	O	Y signal Output
BIASY	1	I	Y D/A Bias-capacitor terminal
COUT	1	O	C signal Output
BIASC	1	I	C D/A Bias-capacitor terminal
COMPOUT	1	O	Composite signal Output
BIASCOMP	1	I	Composite Bias-capacitor terminal
IREF	1	I	Reference-current input
COMP	1	I	Compensation-capacitor terminal
VREF	1	I	Voltage reference
SDRAM Interface			
SDATA[15:0]	16	I/O	SDRAM Data bus.
SADDR[11:0]	12	O	SDRAM Address bus.

Signal Name	#	I/O	Description
SRAS	1	O	SDRAM Row Address Strobe
SCAS	1	O	SDRAM Column Address Strobe
SWE	1	O	SDRAM Write Enable
SDOMU	1	O	SDRAM Data Mask Enable, Upper byte.
SDOML	1	O	SDRAM Data Mask Enable, Lower byte.
SCLK	1	O	SDRAM Clock
SCKE	1	O	SDRAM Clock Enable
SCS1	1	O	SDRAM Chip Select 1
SCS2	1	O	SDRAM Chip Select 2
Device Control:			
RESET*	1	I	Reset, active low
TDI*	1	I	JTAG Data Input. Can be tied high or left floating.
TCK*	1	I	JTAG Clock. Must be tied low for normal operation.
TMS*	1	I	JTAG Test Mode Select Can be tied high or left floating.
TRST*	1	I	JTAG Test Reset, active low. Must be tied low or connected to RESET for normal operations.
TDO*	1	O	JTAG Data Output
Reserved	3		Reserved for Test
VCC / GND	10		Analog supply
VCC / GND	44		Digital supply

* indicates a 5 volt tolerant pin

Claims

1. Audio Visual Circuitry, comprising:

a transport packet parsing circuit for receiving a transport data packet stream,
a processor for initializing said integrated circuit and for processing portions of said data packet stream,
a ROM circuit for storing data,
a RAM circuit for storing data,
an audio decoder circuit for decoding audio portions of said data packet stream,
a video decoder circuit for decoding video portions of said data packet stream,
an NTSC/PAL encoding circuit for encoding video portions of said data packet stream,
an OSD coprocessor circuit for processing OSD portions of said data packets,
a traffic controller circuit for moving portions of said data packet stream between portions of said integrated circuit,
an extension bus interface circuit,
a P1394 interface circuit,
a communication coprocessor circuit,
an address bus connected to said circuits, and
a data bus connected to said circuits.

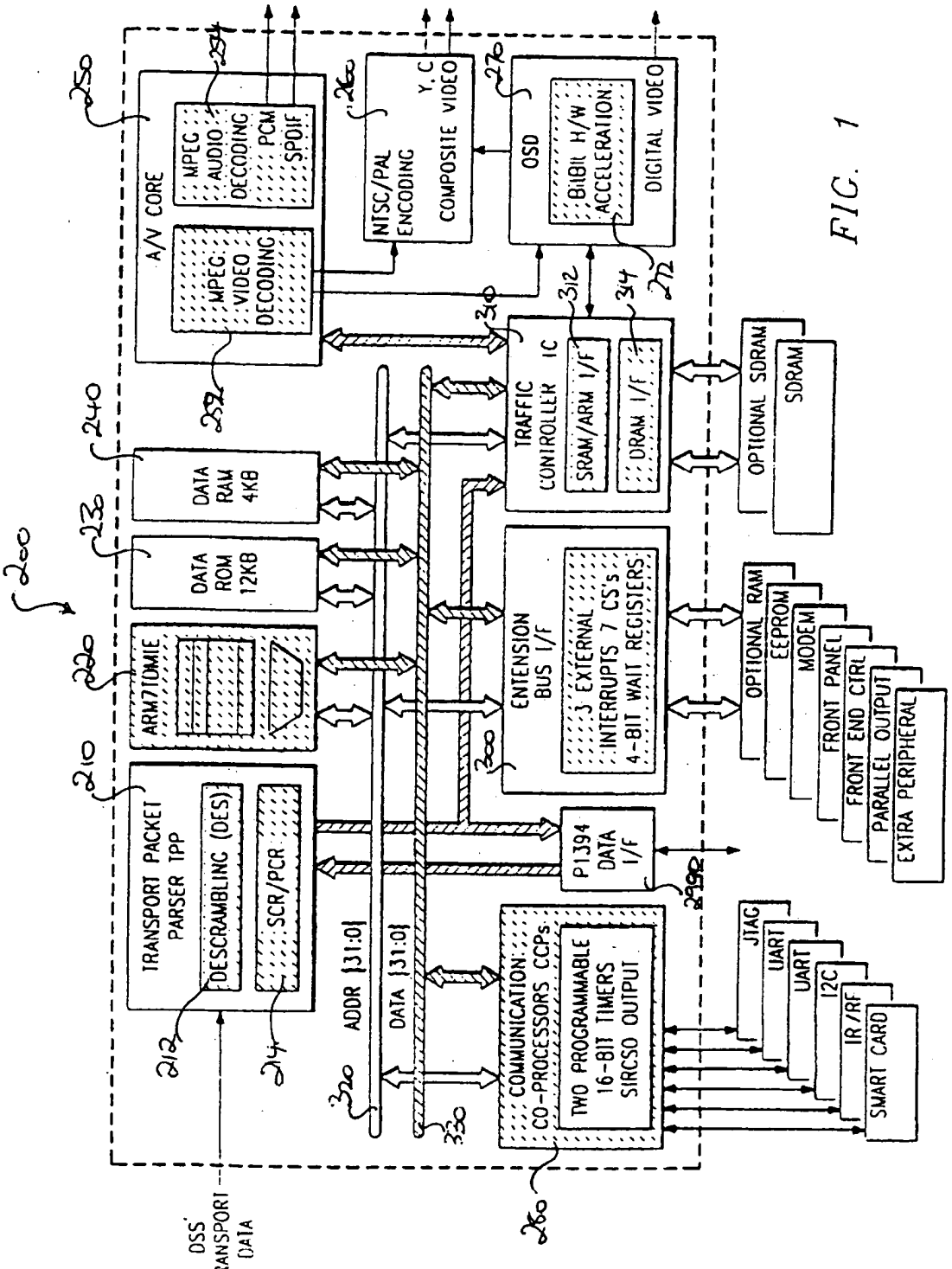

200
210
ARM7TDMIE
TRANSPORT PACKET PARSER TPP
DESCRAMBLING (DES)
SCR/PCR
212
214
220
230
DATA ROM 12KB
240
DATA RAM 4KB
250
A/V CORE
MPEG AUDIO DECODING
PCM
SPDIF
MPEG VIDEO DECODING
252
254
260
NTSC/PAL ENCODING
Y, C COMPOSITE VIDEO
270
OSD
BitBlt H/W ACCELERATION
DIGITAL VIDEO
272
300
EXTENSION BUS I/F
3 EXTERNAL INTERRUPTS
7 CS's
4-BIT WAIT REGISTERS
310
TRAFFIC CONTROLLER IC
312
SRAM/ARM I/F
314
DRAM I/F
320
ADDR [31:0]
DATA [31:0]
330
COMMUNICATION CO-PROCESSORS CCPs
TWO PROGRAMMABLE 16-BIT TIMERS
SIRCSO OUTPUT
280
P1394 DATA I/F
290
OPTIONAL SDRAM
SDRAM
OPTIONAL RAM
EEPROM
MODEM
FRONT PANEL
FRONT END CTRL
PARALLEL OUTPUT
EXTRA PERIPHERAL
JTAG
UART
UART
I2C
IR/RF
SMART CARD
DSS TRANSPORT DATA

FIG. 1

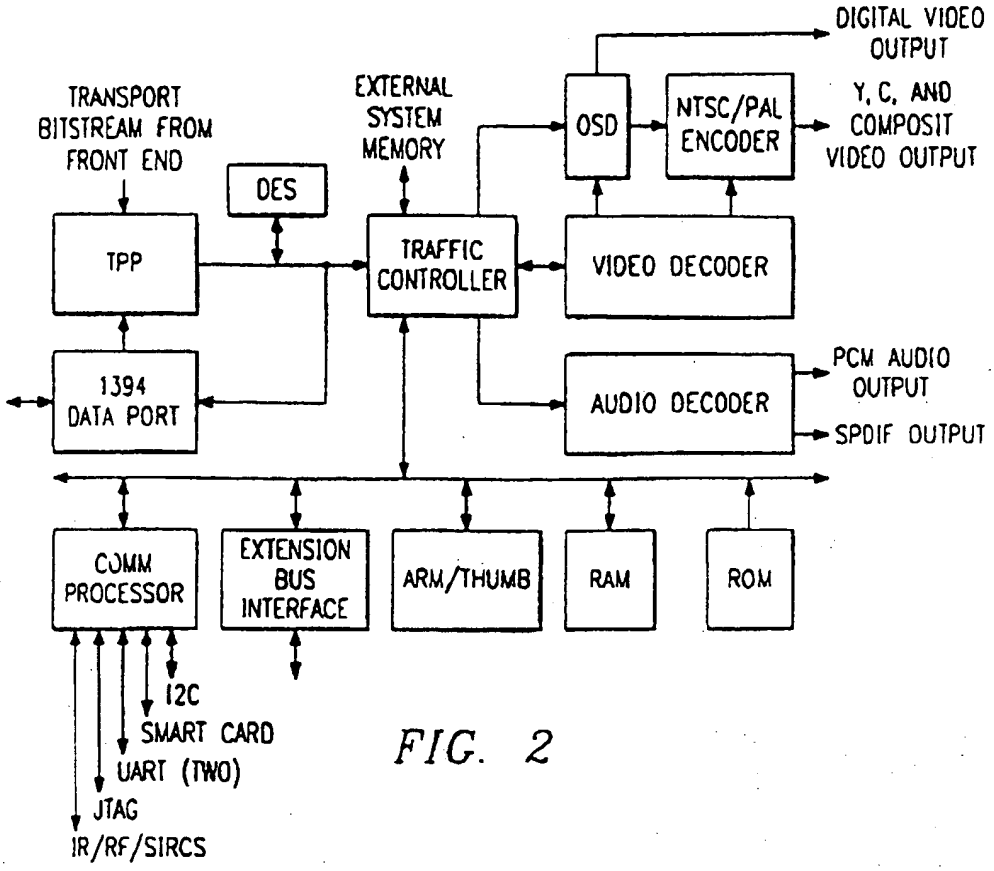

DIGITAL VIDEO OUTPUT
TRANSPORT BITSTREAM FROM FRONT END
EXTERNAL SYSTEM MEMORY
OSD
NTSC/PAL ENCODER
Y, C, AND COMPOSIT VIDEO OUTPUT
DES
TPP
TRAFFIC CONTROLLER
VIDEO DECODER
1394 DATA PORT
PCM AUDIO OUTPUT
AUDIO DECODER
SPDIF OUTPUT
COMM PROCESSOR
EXTENSION BUS INTERFACE
ARM/THUMB
RAM
ROM
I2C
SMART CARD
UART (TWO)
JTAG
IR/RF/SIRCS

FIG. 2

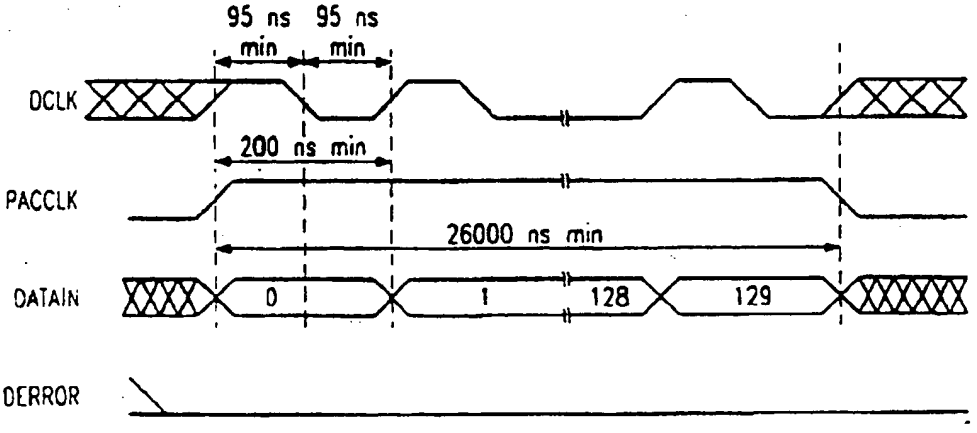

95 ns min
95 ns min
DCLK
200 ns min
PACCLK
26000 ns min
DATAIN
0
1
128
129
DERROR

FIG. 3

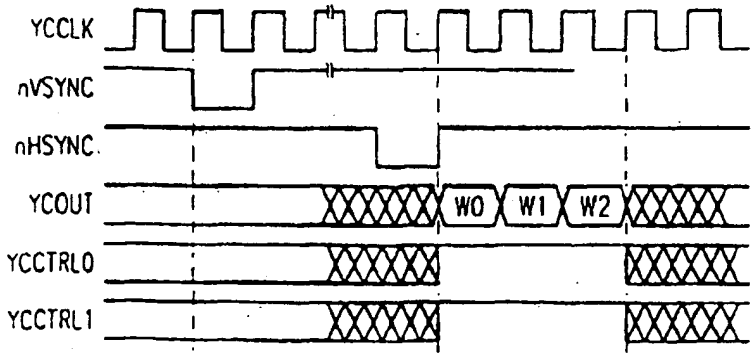


FIG. 4

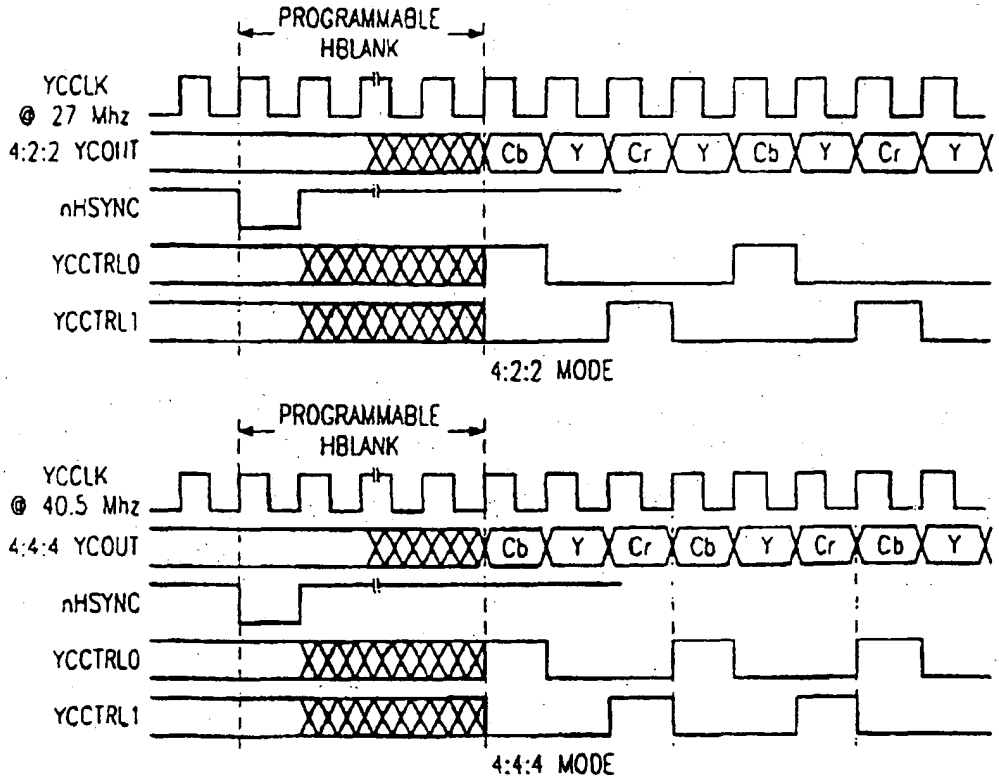


FIG. 5

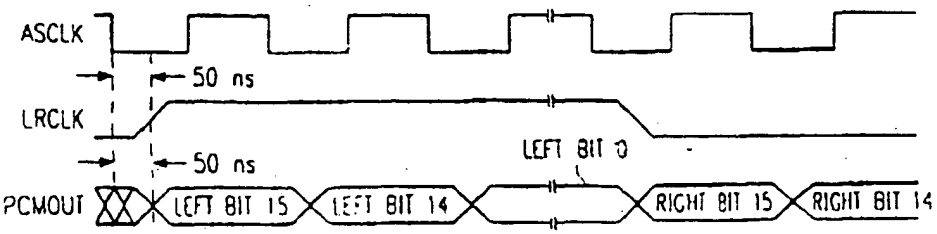
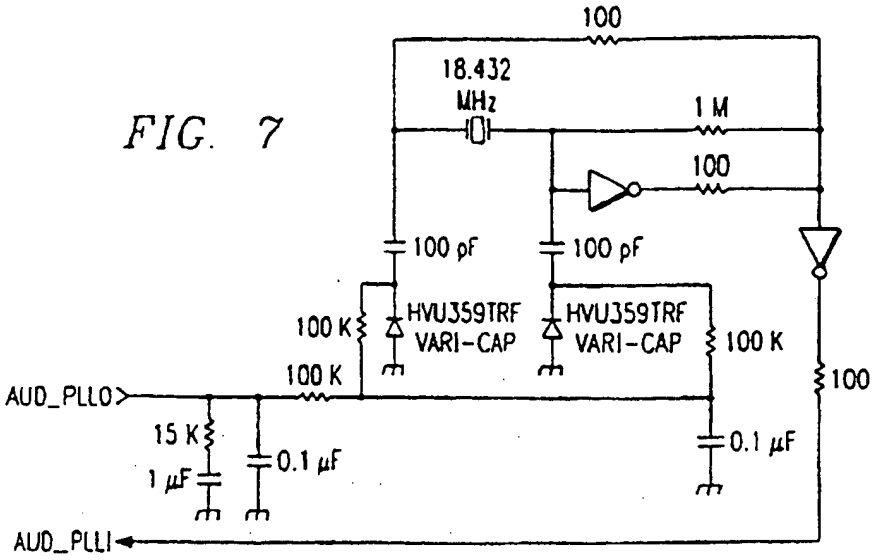


FIG. 6

FIG. 7



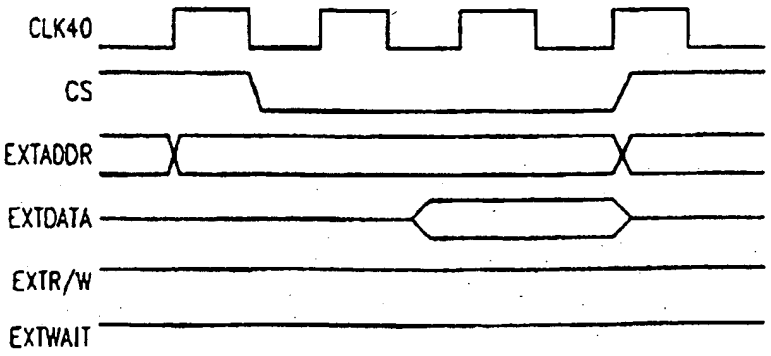


FIG. 8

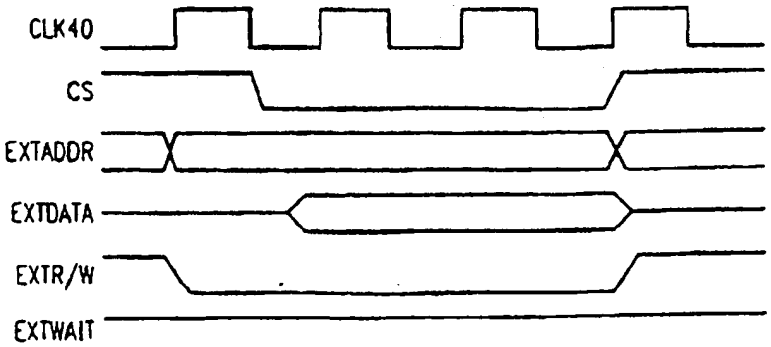


FIG. 9

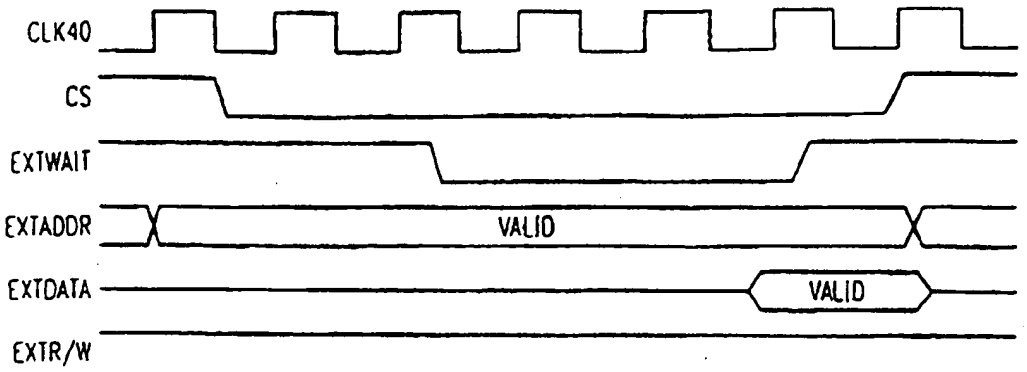


FIG. 10

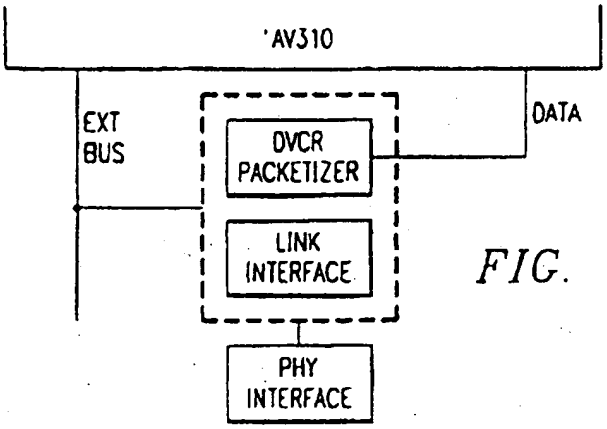


FIG. 11

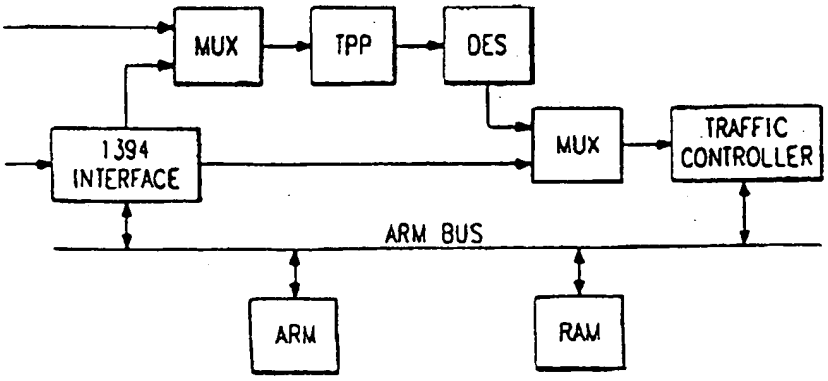


FIG. 12

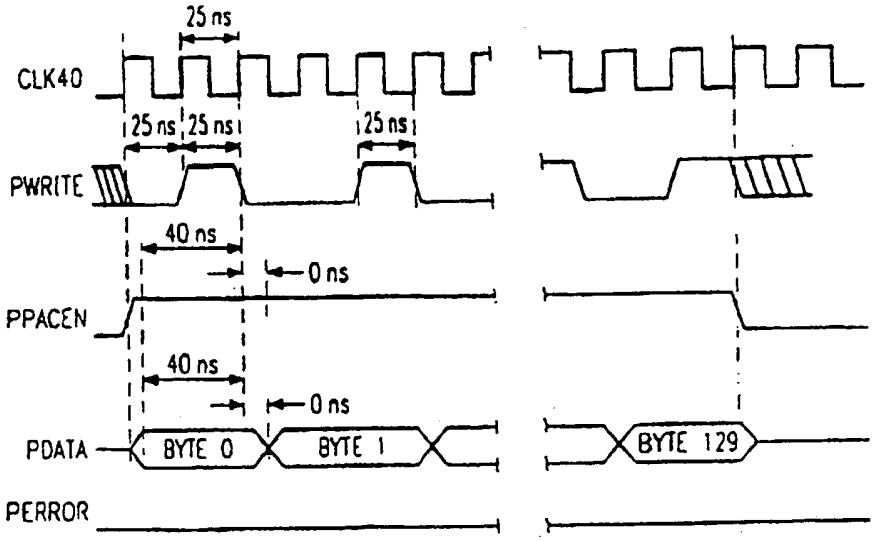

25 ns
CLK40
25 ns
25 ns
25 ns
PWRITE
40 ns
0 ns
PPACEN
40 ns
0 ns
PDATA
BYTE 0
BYTE 1
BYTE 129
PERROR

FIG. 13

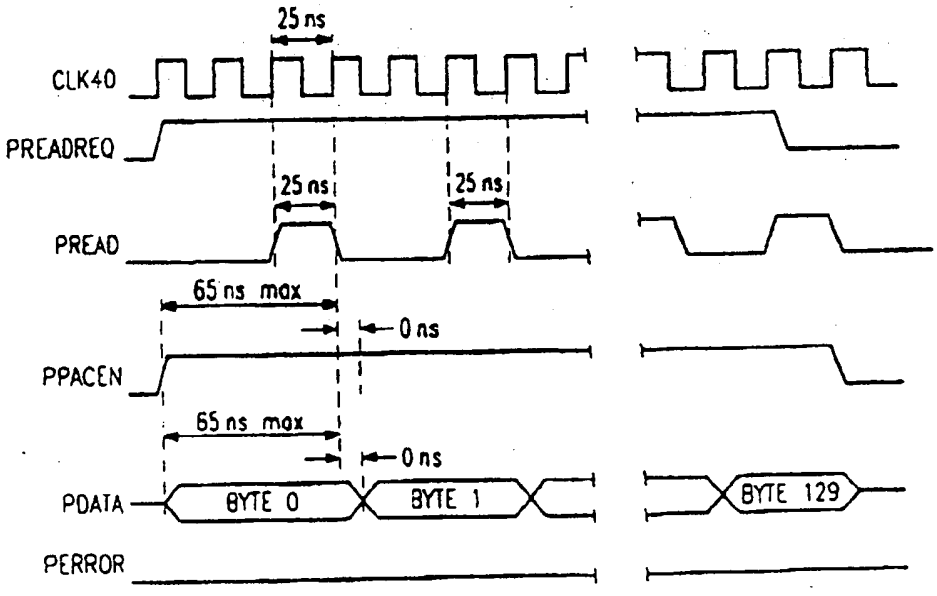

25 ns
CLK40
PREADREQ
25 ns
25 ns
PREAD
65 ns max
0 ns
PPACEN
65 ns max
0 ns
PDATA
BYTE 0
BYTE 1
BYTE 129
PERROR

FIG. 14

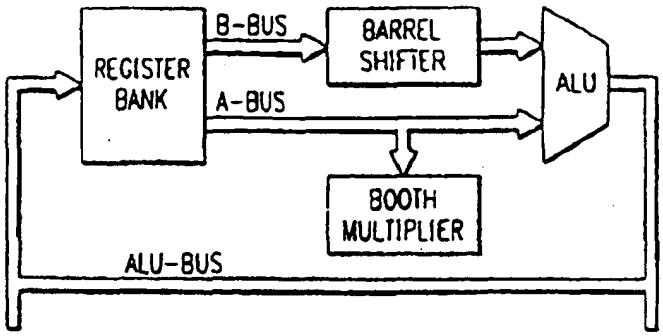

REGISTER BANK
B-BUS
BARREL SHIFTER
ALU
A-BUS
BOOTH MULTIPLIER
ALU-BUS

FIG. 15

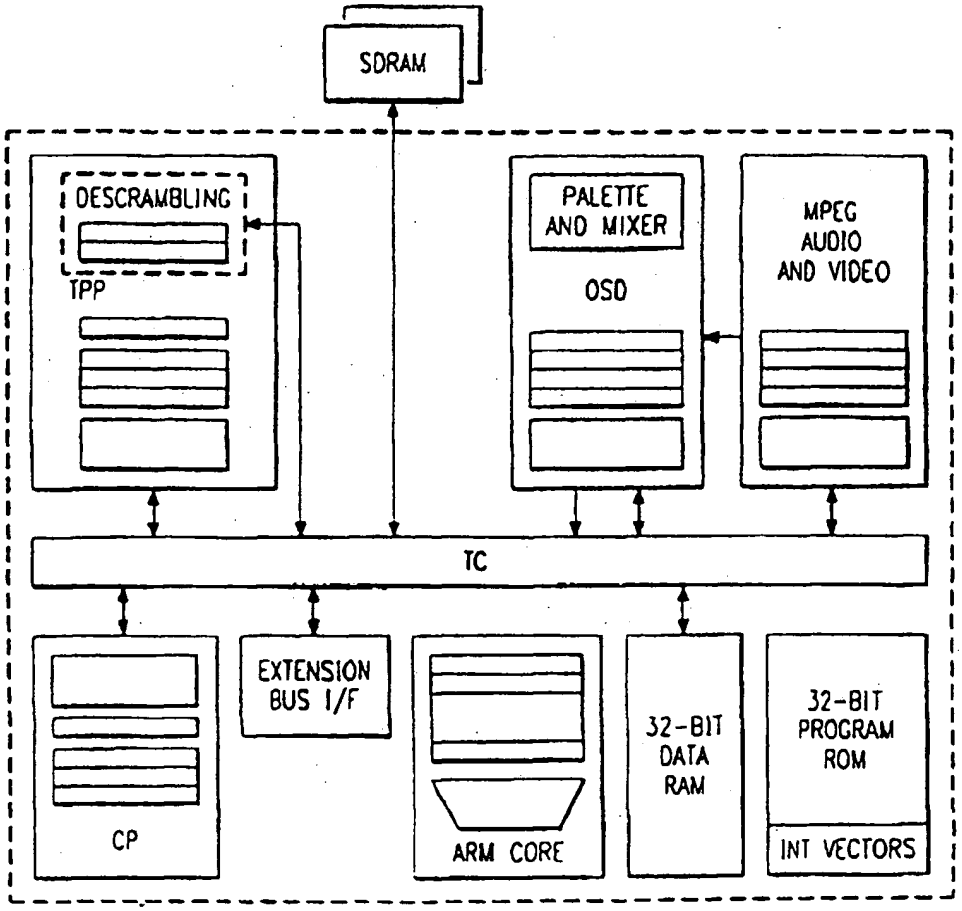

SDRAM
DESCRAMBLING
TPP
PALETTE AND MIXER
OSD
MPEG AUDIO AND VIDEO
TC
CP
EXTENSION BUS I/F
ARM CORE
32-BIT DATA RAM
32-BIT PROGRAM ROM
INT VECTORS

FIG. 16

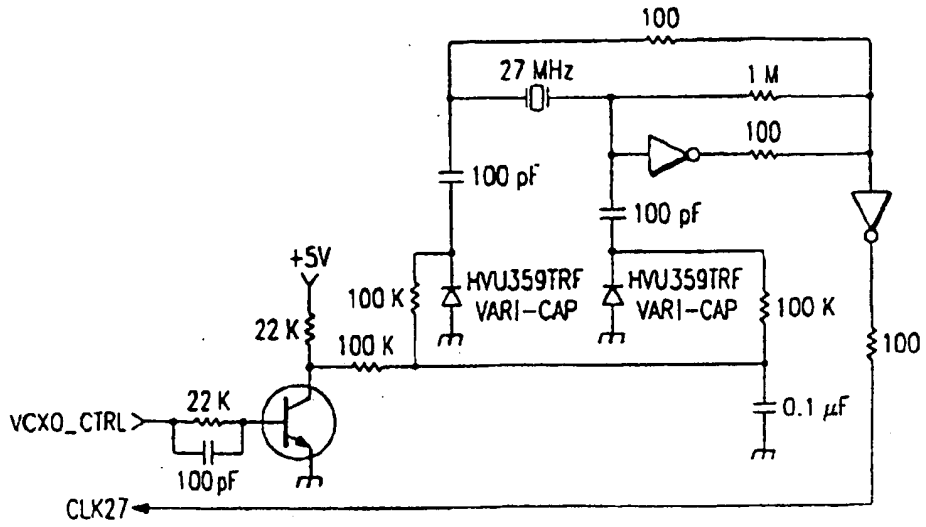

100
27 MHz
1 M
100
100 pF
100 pF
+5V
100 K
HVU359TRF
VARI-CAP
HVU359TRF
VARI-CAP
100 K
22 K
100 K
100
22 K
0.1 μF
VCXO_CTRL
100pF
CLK27

FIG. 17

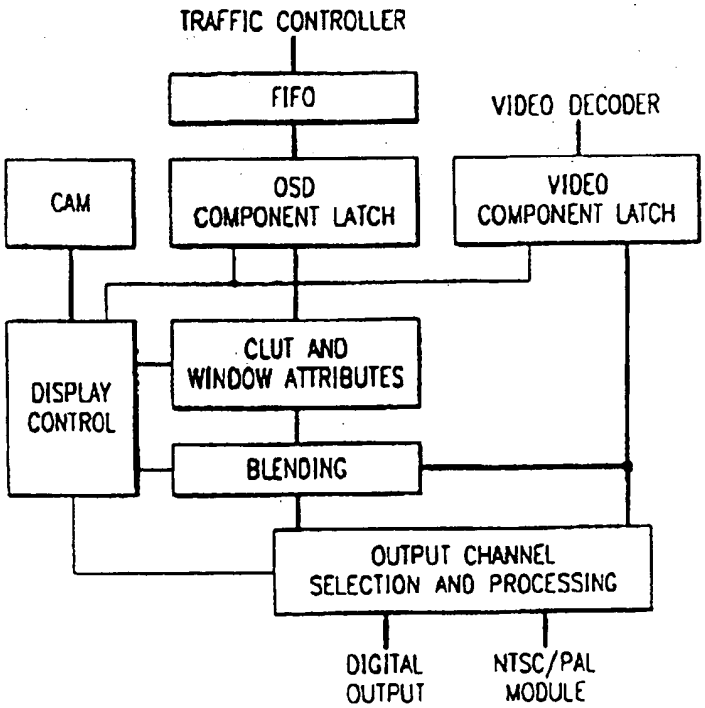

TRAFFIC CONTROLLER
FIFO
VIDEO DECODER
CAM
OSD
COMPONENT LATCH
VIDEO
COMPONENT LATCH
DISPLAY
CONTROL
CLUT AND
WINDOW ATTRIBUTES
BLENDING
OUTPUT CHANNEL
SELECTION AND PROCESSING
DIGITAL
OUTPUT
NTSC/PAL
MODULE

FIG. 18

CH1 NTSC/PAL ENCODER OUTPUTS / CH2 DIGITAL VIDEO OUTPUT	VIDEO ONLY	FULL OSD PICTURE	NON-OVERLAPPED OSD	BOTTOM OF OVERLAPPED OSD
VIDEO ONLY	YES	YES	YES	YES
FULL OSD PICTURE	YES	YES	YES	NO
NON-OVERLAPPED OSD	YES	YES	YES	NO
TOP OF OVERLAPPED OSD	YES	YES	YES	NO

FIG. 19

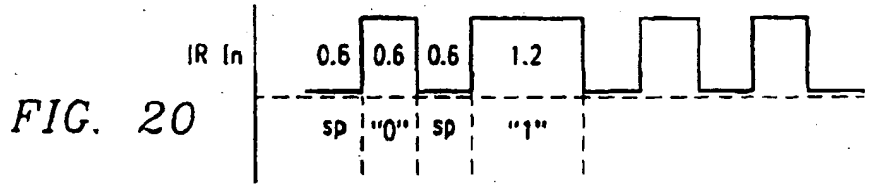


FIG. 20

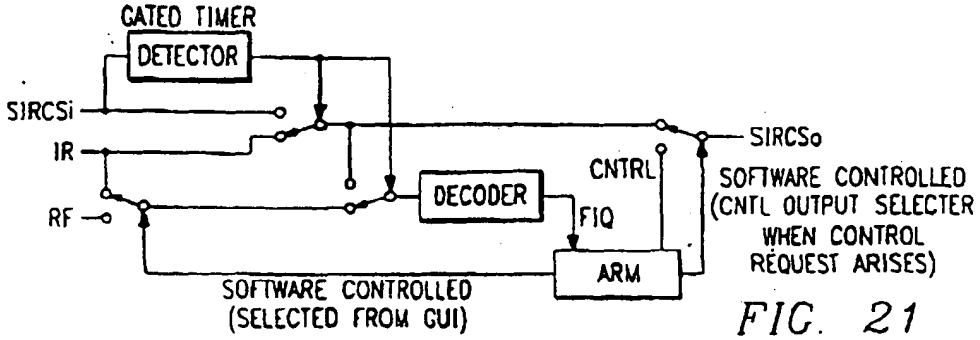


FIG. 21